MCR-74-464
Contract NAS8-29944

Final Report

January 1975

Finite Element Analysis of Mercury Slosh in the Solar Electric Propulsion Stage

(NASA-CR-120597) FINITE ELEMENT ANALYSIS OF MERCURY SLOSH IN THE SOLAR ELECTRIC PROPULSION STAGE Final Report (Martin Marietta Corp.) 136 p HC $5.75 CSCL 21C

N75-15737

Unclas
G3/20 08547

Prepared for:
Marshall Space Flight Center
Huntsville, Alabama 35823

MARTIN MARIETTA

MCR-74-464

"FINITE ELEMENT ANALYSIS OF MERCURY SLOSH IN THE SOLAR ELECTRIC PROPULSION STAGE"

Contract NAS8-29944

(Final Report)

Approved by:

George Morosow

George Morosow
Program Manager

January 1975

Author: J. N. Singh

Prepared for: Marshall Space Flight Center
Huntsville, Alabama 35812

MARTIN MARIETTA CORPORATION
Denver Division
Denver, Colorado 80201

FOREWORD

This report, prepared by the Dynamics and Loads Section, Martin Marietta Corporation, Denver Division, under Contract NAS8-29944, presents the technical approach and the results of a study contract for the dynamic characteristics of a spherical tank/fluid/bladder to be used in Solar Electric Propulsion Stage (SEPS). The study was administered by the National Aeronautics and Space Administrations, George C. Marshall Space Flight Center, Huntsville, Alabama, under the direction of Mr. Frank Bugg, Systems Dynamics Laboratory.

CONTENTS

Figures

ACKNOWLEDGEMENTS

The author appreciates the assistance necessary for the successful completion of the study. Mr. Frank Bugg, of NASA, had many valuable comments and suggestions. The very helpful comments and suggestions provided by Mr. Carl Bodley of the Dynamics and Loads Section, Martin Marietta Corporation, Denver Division, is very deeply appreciated. Thanks are also due to Mr. Peter Abbott, Mr. Richard Wohlen, Mr. Dave Warner, Mr. Wilcomb Benfield, Mr. Roger Philippus, for their effort and assistance. The author also is very deeply appreciative of the helpful guidance provided by Mr. George Morosow, the Program Manager.

INTRODUCTION

The proposed Solar Electric Propulsion Stage (SEPS) will be responsible for the transmittal of Shuttle-based payloads between low orbit and geosynchronous orbit. The SEPS will be a reuseable vehicle powered by accelerated mercury ions. The mercury propellant will be contained in spherical tanks. Propellant orientation within the tank will be controlled by a neoprene expulsion bladder; propellant feed to the ion engines will be maintained by insertion of freon pressurant into the ullage space between the tank and the expulsion bladder. The mass of the mercury will be a very significant portion of the total mass of the spacecraft. It is, therefore, apparent that this high mass fraction cannot be ignored and its dynamic characteristics must be investigated.

This study obtained the equilibrium shapes, vibration modal properties, and mechanical equivalent slosh models for the mercury and bladder for five fill conditions and two acceleration levels.

1. SCOPE

1.1 Purpose

The purpose of this report is to document the investigation performed under contract NAS8-29944, "Finite Element Analysis of Mercury Slosh in the Solar Electric Propulsion Stage".

1.2 Scope

This report documents the dynamic characteristics of the system of spherical tank/hemispherical expulsion bladder/mercury propellant. The static equilibrium shape corresponding to various ullage and gravity configurations was established. The finite-element approach was applied to different fill conditions in different gravity fields to evaluate the lateral sloshing mode shapes and frequencies. The resulting mode shapes and frequencies was used to define a spring-mass mechanical analog that describes the sloshing phenomenon. Computer programs for the equilibrium shape and vibration analysis including the modeling were developed.

1.3 Summary

The static equilibrium shapes of the neoprene bladder have been established corresponding to various ullage and gravity configurations under specified boundary conditions. The hemispherical bladder is taken to be attached at the diametral plane of the sphere with zero relative slope. With these shapes, the spherical tank with bladder and mercury has been modeled as an assemblage of finite-elements. The properties of these elements have then been calculated using a linear displacement field. The dynamic characteristics were obtained to be used to define a mechanical analog which will reproduce the sloshing phenomenon of the system. The computer programs for the static free surface and vibration analysis have been checked out.

2. TECHNICAL APPROACH

The problem was approached in two distinct steps.

(1) Establish the static equilibrium shape.

(2) Using the free surface shape from (1), model the system as an assemblage of finite elements and obtain the mode shapes and frequencies. Further define a mechanical analog for the system.

2.1 Static Equilibrium Shape - The method comprises of determining the total energy of the system in terms of the variables involved and minimizing the total energy for the stable equilibrium state under the given constraints. This is outlined below. However, it is to be pointed out here that the idealized case here is for an infinitely long cylinder and the feeling being that the static free surface for this case will be close to the actual case of spherical tank.

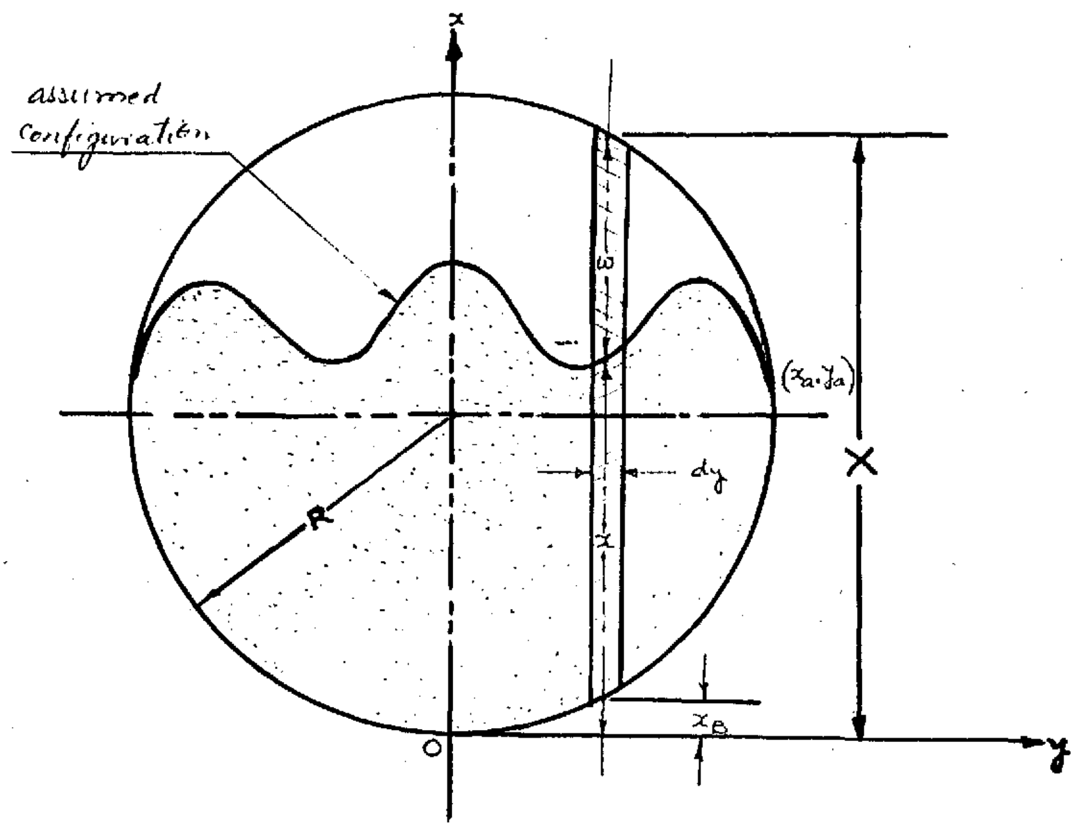


Figure 1

The equation for the circle (with origin at center)

$$\bar{x}^2 + \bar{y}^2 = R^2 \tag{1}$$

Taking the origin at bottom point

$$X = \pm [R^2 - y^2]^{1/2} + R \tag{2}$$

Introducing w(y) in the expression

$$x(y) = X(y) - w(y) = \pm[R^2 - y^2]^{1/2} + R - w(y) \tag{4}$$

The plus sign represents the upper point of the sphere and the negative sign the bottom and hence,

$$x(y) = X(y) - w(y) = [R^2 - y^2]^{1/2} + R - w(y) \tag{5}$$

An assumption is made here that w(y) may be expressed in terms of a power series:

$$w(y) = \sum_{i=0}^{N} a_i y^i \tag{6}$$

Substituting for w(y) from equation (6) in equation (5),

$$x(y) = [R^2 - y^2]^{1/2} + R - \sum_{i=0}^{N} a_i y^i \tag{7}$$

The stretching in the membrane strip of length s measured along the curved shape is $\tilde{u}(s)$. It is assumed that

$$\tilde{u}(s) = \sigma s \tag{8}$$

Therefore,

$$\frac{\partial \tilde{u}}{\partial s} = \sigma \tag{9}$$

That is, σ is the strain in the membrane.

The total potential of the assumed configuration will now be calculated. They are:

(i) Bladder strain energy in bending:

$$dV_B = \frac{1}{2} \frac{EI\,ds}{\rho^2} \tag{10}$$

where: dV_B = strain energy in bending

E = modulus of elasticity for the bladder

I = moment of inertia

dS = length

ρ = radius of curvature

Further

$$\frac{1}{\rho} = \frac{d^2x/dy^2}{\left[1+\left(\frac{dx}{dy}\right)^2\right]^{3/2}} \tag{11}$$

Therefore, equation (10) can be rewritten as

$$dV_B = \frac{1}{2} EI \frac{\left(d^2x/dy^2\right)^2}{\left[1+\left(\frac{dx}{dy}\right)^2\right]^3} dS \tag{12}$$

It can easily be seen that

$$dS = \left[\frac{1}{1+\left(\frac{dx}{dy}\right)^2}\right]^{-1/2} dy \tag{13}$$

Substituting from (13) in (12)

$$dV_B = \frac{1}{2} EI \frac{\left(\frac{d^2x}{dy^2}\right)^2}{\left[1+\left(\frac{dx}{dy}\right)^2\right]^{5/2}} dy \tag{14}$$

Therefore,

$$V_B = \frac{1}{24} E h^3 \int_0^{y_a} \frac{\left(\frac{d^2x}{dy^2}\right)^2}{\left[1+\left(\frac{dx}{dy}\right)^2\right]^{5/2}} dy \tag{15}$$

where: h = thickness of the bladder

(ii) The strain energy due to membrane action:

$$dV_M = \tfrac{1}{2}(\text{stress})\cdot(\text{strain})\ h\ dS$$

Resulting in

$$V_M = \frac{1}{2} E h \delta^2 \int_0^{y_a} \left[1+\left(\frac{dx}{dy}\right)^2\right]^{1/2} dy \tag{16}$$

where V_M = strain energy due to membrane action.

(iii) Fluid gravitational potential energy:

$$dV_g = (x - x_B)\,dy \cdot 1 \cdot \rho \cdot g \cdot \{x_B + (x - x_B)/2\} \tag{17}$$

Simplifying,

$$V_g = \tfrac{1}{2}\rho g \int_0^{y_a} x^2 dy - \tfrac{1}{2}\rho g \int_0^{y_a} x_B^2 dy \tag{18}$$

where: ρ = mass density of the fluid
g = acceleration due to gravity
x_B = as shown in Figure 1
V_g = gravitational potential energy

(iv) Work done by ullage pressure:

$$dW = p.dV \tag{19}$$

where: p = ullage pressure
Therefore, virtual work by ullage pressure due to virtual change in w,

$$\delta W = p \int_0^{y_a} \delta w . dy \tag{20}$$

where: δw = virtual change w
δW = virtual work

Substituting for δw from (6) in (20),

$$\delta W = p \sum_{i=0}^{N} \delta a_i \int_0^{y_a} y^i dy \tag{21}$$

as

$$W = W\,(a_0,\ a_1,\ a_2,\ \ldots\ldots\ldots,\ a_N, \sigma) \tag{22}$$

$$\delta W = \sum_{i=0}^{N} \frac{\partial W}{\partial a_i}\delta a_i + \frac{\partial W}{\partial \sigma}\delta\sigma \tag{23}$$

The comparison of equations (21) and (23) reveals,

$$\frac{\partial W}{\partial a_i} = p \int_0^{y_a} y^i \, dy \tag{24}$$

for i = 0, 1, 2, ..., N

and

$$\frac{\partial W}{\partial \sigma} = 0 \tag{25}$$

In the present case, the system may be treated as conservative for a particular time, and therefore

$$V = -W \tag{26}$$

where: V = is the total potential energy of the system
W = work function

Substituting in equation (24),

$$\frac{\partial V}{\partial a_i} = -p \int_0^{y_a} y^i \, dy \tag{27a}$$

and

$$\frac{\partial V}{\partial \sigma} = 0 \tag{27b}$$

under the boundary conditions of

$$\begin{aligned} \omega(y_a) &= 0 \\ \omega'(y_a) &= 0 \\ \omega'(0) &= 0 \\ \int_0^{y_a} \omega \, dy &= K \end{aligned} \tag{28}$$

where: 2K = given ullage volume

and

$$V = V_B + V_M + V_g \tag{29}$$

Assembling the expression

$$V = \frac{Eh^3}{24} \int_0^{y_a} \frac{d^2x/dy^2}{\left[1+\left(\frac{dx}{dy}\right)^2\right]^{5/2}} dy + \frac{Eh\sigma^2}{2} \int_0 \left[1+\left(\frac{dx}{dy}\right)^2\right]^{1/2} dy + \frac{\rho g}{2} \int_0^{y_a} \left(x^2 - x_B^2\right) dy \tag{30}$$

The equation (30) has to be expressed in terms of a_i's and .

$$x(y) = X(y) - w(y) = \left(R^2 - y^2\right)^{1/2} + R - \sum_{i=0}^{N} a_i\, y^i \tag{31}$$

$$x' = \frac{dx}{dy} = -y\left(R^2 - y^2\right)^{-1/2} - \sum_{i=0}^{N} i\, a_i\, y^{i-1} \tag{32}$$

$$x'' = \frac{d^2x}{dy^2} = -\left(R^2 - y^2\right)^{-1/2} - y^2\left(R^2 - y^2\right)^{-3/2} - \sum_{i=0}^{N} i(i-1)\, a_i\, y^{i-2} \tag{33}$$

Substituting from (31), (32) and (33) in equation (30) and differentiating with respect to a_i's and b respectively and comparing with (27a) and (27b),

$$\begin{aligned}\frac{\partial V}{\partial a_i} &= -p\int_0^{y_a} y^i\, dy \\ &= \frac{Eh^3}{12}\int_0^{y_a} \frac{\left(\{1+(x')^2\}\{x''\}\{-i(i-1)y^{i-2}\} - \{5/2\}\{x''\}^2\{x'\}\{-i\,y^{i-1}\}\right)}{\left(1+\{x'\}^2\right)^{7/2}}\, dy \\ &\quad + \frac{Ehb^2}{2}\int_0^{y_a}\left[\{1+(x')^2\}^{-1/2}\{x'\}\{-i\,y^{i-1}\}\right] dy \\ &\quad + \rho g\int_0^{y_a}\left[\{x\}\{-y^i\}\right] dy\end{aligned} \tag{34a}$$

and

$$\frac{\partial V}{\partial b} = 0 = Ehb\int_0^{y_a}\left[1+\{x'\}^2\right] dy \tag{34b}$$

for i = 0, 1, 2, ..., N

Under the auxiliary conditions of

$$\begin{aligned} w(y_a) &= 0 \\ w'(y_a) &= 0 \\ w'(0) &= 0 \\ \int_0^{y_a} w\, dy &= K \end{aligned} \tag{35}$$

Finally, the equations of the system are

$$\frac{Eh^3}{12}\int_0^{\gamma a}\frac{[\{1+(x')^2\}\{x''\}\{-i(i-1)y^{i-2}\}-\{5/2\}\{x''\}^2\{x'\}\{-iy^{i-1}\}]}{[1+\{x'\}^2]^{7/2}}dy$$
$$+\frac{Eh\sigma^2}{2}\int_0^{\gamma a}[\{1+(x')^2\}^{-1/2}\{-x'\}\{-iy^{i-1}\}]\,dy$$
$$+\rho g\int_0^{\gamma a}[\{x\}\{-y^i\}]dy+p\int_0^{\gamma a}y^i dy=0 \tag{36}$$

and

$$Eh\sigma\int_0^{\gamma a}[1+\{x'\}^2]\,dy=0 \tag{37}$$

for i = 0, 1, 2, ..., N

under the conditions

$$\begin{aligned} w(\gamma a)&=0\\ w'(\gamma a)&=0\\ w'(0)&=0\\ \int_0^{\gamma a}w\,dy&=k \end{aligned} \tag{38}$$

Thus,

$$F=F(a_0,a_1,\ldots,a_N,\sigma)=F(q_1,q_2,\ldots,q_{N+2}) \tag{39}$$

In general, the equations can be written as,

$$F_n(q_i)=\frac{Eh^3}{12}\int_0^{\gamma a}\frac{[\{1+(x')^2\}\{x''\}\{-n(n-1)(n-2)y^{n-3}\}-\{5/2\}\{x''\}^2\{x'\}\{-(n-1)y^{n-2}\}]}{[1+\{x'\}^2]^{7/2}}dy$$
$$+\frac{Eh\sigma^2}{2}\int_0^{\gamma a}[\{1+(x')^2\}^{-1/2}\{-x'\}\{-(n-1)y^{n-2}\}]dy$$
$$+\rho g\int_0^{\gamma a}[\{x\}\{-y^{n-1}\}]dy+p\int_0^{\gamma a}y^{n-1}dy=0 \tag{40a}$$

for n = 1, 2, ..., N+1

$$F_n(q_i)=Eh\sigma\int^{\gamma a}[1+(x')^2]^{1/2}dy=0 \tag{40b}$$

for n = N+2

under the auxiliary conditions

$$
\begin{aligned}
\omega(y_a) &= 0 \\
\omega'(y_a) &= 0 \\
\omega'(0) &= 0 \\
\int_0^{y_a} \omega \, dy &= K
\end{aligned}
\qquad (40c)
$$

The system of equations (40), i.e., (40a), (40b) are to be solved under conditions of (40c) for q_i with iterative approach. Denoting $(F_n)_s$, where s stands for iteration step, $(F_n)_o$ stands for initial assumption,

$$
(F_n)_1 = (F_n)_o + \frac{\partial}{\partial q_m}(F_n)_o \, \Delta q_m \qquad (41)
$$

In general, therefore,

$$
(F_n)_{s+1} = (F_n)_s + \frac{\partial}{\partial q_m}(F_n)_s \, \Delta q_m \qquad (42)
$$

where n = 1, 2, ..., N+2 and m = 1, 2, ..., N+2 (separately). Rewritting the equations (42) in matrix form:

$$
\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ \vdots \\ \vdots \\ F_{N+2} \end{Bmatrix}_{s+1}
=
\begin{bmatrix}
\partial F_1/\partial q_1 & \partial F_1/\partial q_2 & \partial F_1/\partial q_3 & \cdots & \partial F_1/\partial q_{N+2} \\
\partial F_2/\partial q_1 & \partial F_2/\partial q_2 & \partial F_2/\partial q_3 & \cdots & \partial F_2/\partial q_{N+2} \\
\vdots & \vdots & \vdots & & \vdots \\
\vdots & \vdots & \vdots & \cdots & \vdots \\
\frac{\partial F_{N+2}}{\partial q_1} & \frac{\partial F_{N+2}}{\partial q_2} & \frac{\partial F_{N+2}}{\partial q_3} & \cdots & \frac{\partial F_{N+2}}{\partial q_{N+2}}
\end{bmatrix}_s
\begin{Bmatrix} \Delta q_1 \\ \Delta q_2 \\ \Delta q_3 \\ \vdots \\ \vdots \\ \Delta q_{N+2} \end{Bmatrix}
\qquad (43)
$$

The various terms involved in the matrix are as follows:

$$
\begin{aligned}
\frac{\partial F_n}{\partial q_m} = \frac{Eh^3}{12}\int_0^{y_a} \Big[\Big(\{1+(x')^2\}^{-7/2} \Big) \Big([\{(n-1)(n-2)y^{n-3}\}] [\{1+(x')^2\} \tfrac{\partial}{\partial a_m}(x') \\
+ (x'')\{2(x')\tfrac{\partial}{\partial a_m}(x')\}] - (5/2)[-(n-1)y^{n-2}\{(x'')^2 \tfrac{\partial}{\partial a_m}(x') + 2(x')(x'')\tfrac{\partial}{\partial a_m}(x'')\}] \Big) \qquad (44a) \\
+ \Big([1+(x')^2][x''][(n-1)(n-2)y^{n-3}] - (5/2)(x'')^2(x')[-(n-1)y^{n-2}] \Big)
\end{aligned}
$$

$$\left(\left[-\tfrac{7}{2}\right]\left[1+(x')^2\right]^{-9/2}\left[2(x')\right]\left[\frac{\partial}{\partial a_m}(x')\right]\right)\Big]\,dy$$
$$+\frac{Ehb^2}{2}\int_0^{y_a}\Big[\left(\left\{1+(x')^2\right\}^{-1/2}\left\{\frac{\partial}{\partial a_m}(x')\right\}+\left(-\tfrac{1}{2}\right)\left\{1+(x')^2\right\}^{-3/2}\left\{2(x')\frac{\partial}{\partial a_m}(x')\right\}\{x'\}\right)$$
$$\left(-\{n-1\}y^{n-2}\right)\Big]\,dy+\rho g\int_0^{y_a}\left(\left[\frac{\partial}{\partial a_m}(x)\right]\left[-y^{n-1}\right]\right)dy$$

for n = 1, 2, ..., N+1 and m = 1, 2, ..., N+1 (separately)

and

$$\frac{\partial F_n}{\partial q_m} = Ehb\int_0^{y_a}\left(\left\{1+(x')^2\right\}^{-1/2}\{x'\}\left\{-(n-1)y^{n-2}\right\}\right)dy \qquad (44b)$$

for n = 1, 2, ..., N+1 and m = N+2 (separately)

Also

$$\frac{\partial F_n}{\partial q_m} = Ehb\int_0^{y_a}\left(\{1/2\}\left\{1+(x')^2\right\}^{-1/2}\left\{2(x')\frac{\partial}{\partial a_m}(x')\right\}\right)dy \qquad (44c)$$

for n = N+2 and m = 1, 2, ..., N+1

and

$$\frac{\partial F_n}{\partial q_m} = Eh\int_0^{y_a}\left(1+\{x'\}^2\right)^{1/2}dy \qquad (44d)$$

for n = N+2 and m = N+2

Recalling equation (13), the arc length can be written as

$$s = \int_0^{y_a}\left(\frac{1.0}{1+(x')^2}\right)^{-1/2}dy \qquad (45)$$

Assuming the bladder to be attached at the diametral plane, the original length of the arc is

$$l = \pi R \qquad (46)$$

Therefore, b can, now, be defined as

$$b = \frac{2s-l}{l} \qquad (47)$$

Thus, the system of equations (40a), (40b), (40c) and (47) are to be solved simultaneously with an iterative approach. This is accomplished in the computer program.

2.2 Vibration Analysis - Vibration analysis of structures having a large number of degrees of freedom is an ever present problem. Digital computer oriented techniques are primarily restricted by core size and computer time. Consequently, economically feasible eigenvalue/eigenvector techniques are needed for large size structural systems.

However, for a large size structural system, if one is interested in a relatively small number of modes, Rayleigh-Ritz technique appears to have the advantage over the others. In this method, essentially a large problem is reduced to a smaller problem for the range of interest only. The solution is obtained as a linear combination of assumed linearly independent mode shapes.

The method consists in using the assumed mode shapes initially, which reduce the number of generalized coordinates used and then, by an iterative procedure, these modes are improved until they converge to the normal vibration modes of the structure.

2.2.1 Rayleigh-Ritz Technique - The iterative Rayleigh-Ritz method(1) is used to calculate the mode shapes and frequencies of the system. This method is based on repeated application of the well known Rayleigh-Ritz technique using improved mode shapes for each iteration. This technique reduces the size of the system without degrading accuracy in the desired frequency range. The technique is briefly described here.

For a discrete coordinate model of a structure having n degrees of freedom, the equations of motion can be written as

$$[M]\{\ddot{x}\} + [K]\{x\} = \{0\} \tag{48}$$

where:

$\{x\} = \{x(t)\}$ vector of discrete coordinate displacements,

$[M]$ = mass matrix

$[K]$ = stiffness matrix

If a solution of the type $\{x\} = \{x\} e^{i\omega t}$, implying a simple harmonic motion is assumed, equations (48) can be written as

$$\left([K] - \omega^2[M]\right)\{x\} = \{0\} \tag{49}$$

Equation (49) is recognized as a matrix eigenvalue problem of order n, whose eigenvectors $\{y\}$ are the mode shapes and whose eigenvalues $\lceil \omega^2 \rfloor$ are the frequencies. A complete sequence of trial vectors

$$\{z\}_1, \{z\}_2, \{z\}_3, ----------\{z\}_n \tag{50}$$

which are linearly independent, is assumed. The displacement $\{x\}$ is then expressed as a linear sum of the first "m" trial vector, that is,

$$\underset{(n\times 1)}{\{x\}} = \underset{(n\times m)}{[z]}\underset{(m\times 1)}{\{q\}} \tag{51}$$

Substitution of equation (51) into (49) and for multiplying by $[Z]^T$ gives

$$\left([\bar{K}] - \omega^2[\bar{M}]\right)\{q\} = \{0\} \tag{52}$$

where:

$$[\bar{K}] = [z]^T[K][z] \tag{53}$$

and

$$[\bar{M}] = [z]^T[M][z] \tag{54}$$

Equation (52) is a matrix eigenvalue problem of reduced order "m" whose eigenvectors are $[y^*]$ and eigenvalues are $\lceil \omega^2 \rfloor$. The solution of equation (52) has the form

$$\{q\} = [y^*]\{q^*\} \tag{55}$$

where: $\{q^*\}$ is the normalized coordinate vector. The eigenvalues, $\lceil \omega^2 \rfloor$, approximate the first "m" eigenvalues of the original structure. The associated approximate eigenvectors $[y]$ of the original structure are obtained by substitution of equation (55) into (51), yielding

$$\underset{(n\times 1)}{\{x\}} = \underset{(n\times m)}{[z]}\,\underset{(m\times m)}{[y^*]}\,\underset{(m\times 1)}{\{q^*\}}$$

or

$$\{x\} = [y]\{q^*\} \tag{56}$$

where:

$$[y] = [z][y^*] \tag{57}$$

The accuracy of the mode shapes $[y]$ and frequencies $[\omega^2]$ obtained depends entirely upon the trial vector $[z]$. If $[z]$ contains the true modal patterns, then the eigensolution for $[y]$ and $[\omega^2]$ are exact. However, in general, that is not the case. Exact results can be obtained for the first "m" modes of the structure if the trial vectors $[z]$ do not have any contribution from modes higher than "m". Thus, an improved set of trial vectors can be calculated by suppressing the contribution of higher modes in approximate mode shapes. The procedure for suppressing the contribution of the higher modes is well known; in fact, it is the basis of the Power or Stodola-Vianello' matrix iteration method(2) of modal analysis. Here, however, the method is applied to all modes simultaneously and is given as,

$$[K][z] = [M][y] \tag{58}$$

The solution is carried out for $[Z]$, which is then used to repeat equations (52) through (58). The cycle can be repeated until all the mode shapes $[Y]$ and frequencies ω^2 have converged to within a prescribed tolerance. Convergence is assured because the technique is equivalent to a power iteration applied simultaneously to all modes. Thus, the convergence theorems associated with the power method are directly applicable. The role of the eigensolution (equation (52) is to prevent all modes from converging on the lowest mode.

Associated with the iterative Rayleigh-Ritz technique are parameters that affect the convergence and hence computer time which will be briefly discussed here. They are:

(i) the initial mode shapes assumed to start the iteration process,

(ii) the number of modes used,

(iii) the repression of higher modes, and

(iv) shifting.

(i) Initially Assumed Mode Shapes - The choice of initial mode shapes plays a very important role in the success of the technique. Inherent with the initial mode shape selection are two basic problems: (1) modes may be missed, and (2) the triple product $[\bar{M}] = [Z]^T[M][Z]$ may be ill-conditioned if the columns of $[Z]$ are not sufficiently independent. It does not appear that there is a way to guarantee that the above two conditions will be met with any selection of $[Z]$, however, the chance of them occuring can be minimized with some judicious selection of the vectors. Without proof or discussion, it is to be pointed out here that if the elements of the vector or of matrix $[Z]$ are randomly generated, it has been found that the chances of the above two conditions being violated is very remote.

(ii) Number of Modes Used - An increase in the number of modes used will, in general, decrease the number of iterations required for convergence. However, if more modes are used, the computer time for each iteration will increase because of the increase in sizes of the matrices used; hence, there is a tradeoff.

(iii) Repression of Higher Modes - As pointed out earlier, exact results can be obtained for the first "m" modes of the structure if the trial vectors in $[Z]$ do not contain any contribution from modes higher than "m". Generalizing, it can be said that an improved set of trial vectors can be calculated by suppressing the contribution from the higher modes in the approximate mode shapes at each step. This is achieved as follows.

$$[Z]_j = [K]^{-1}[M][Z]_{j-1} \tag{59}$$

The subscript j denotes the iteration number. If this iteration is repeated sufficient number of times, modes corresponding to the lowest frequency will be reached. If this iteration is repeated too many times, the mode will repeat itself in one or more columns of $[Z]$ and will render $[Z]^T[M][Z]$ to be ill-conditioned.

Its use here is not to converge to a mode but just to repress the higher modes and, hence, just a one time application is advisable.

However, in this case $[K]^{-1}$ is required for which $[K]$ has to be non-singular. Thus, the technique can be applied only if $[K]$ is not singular or has been made such with some technique as described next.

(iv) <u>Shifting</u> - Shifting is an useful technique to speed the convergence of modes whose eigenvalues are close to the shift value. As an additional benefit of shifting process is the conversion of the stiffness matrix (in case of a free-free structure) from singular to a non-singular matrix. The method is as follows.

The eigenvalue problem is

$$[K][\Phi] = [M][\Phi][\omega^2] \tag{60}$$

where: different quantities carry their usual meaning.
Also, it has to be noticed that $[K]$ may be singular. To introduce the shift value, λ_s , the following operation is performed. The quantity, $\lambda_s[M][\Phi]$ is subtracted from both sides of equation (60). Thus

$$\left([K] - \lambda_s[M]\right)[\Phi] = [M][\Phi]\left([\omega^2] - \lambda_s[I]\right) \tag{61}$$

By definition

$$[\hat{K}] = [K] - \lambda_s[M] \tag{62}$$

and

$$[\Omega^2] = [\omega^2] - \lambda_s[I] \tag{63}$$

Therefore, final equation is

$$[\hat{K}][\Phi] = [M][\Phi][\Omega^2] \tag{64}$$

This is now the eigen-problem to be solved rather than (60). It is to be noticed that $[\hat{K}]$ is non-singular even if $[K]$ was not.

The eigenvalues of the original system are easily obtained as

$$\omega_i^2 = \Omega_i^2 + \lambda_s \tag{65}$$

The convergence will be to the lowest absolute value of Ω^2. Thus, shifting by a value, λ_s, the eigenvalues, ω^2, around this shift point are converged to first.

Some general remarks on <u>Shift</u>:

(a) Analysis of a Free-Structure - Because a free structure has a singular stiffness matrix, the solution of the simultaneous equations in the iteration loop is not possible. However, the shift technique alleviates the problem.

(b) Specific Frequency Range - When a shift value is used, the modes with eigenvalues closest to the shift value will converge first, which enables one to obtain the modes in the desired frequency range only.

(c) Large number of modes - By repeated use of different shift values, any number of modes can be obtained.

(d) The following observations are made without discussion.

(i) If the lowest eigenvalues in the range ω_1^2, ω_2^2, ..., ω_i^2, are needed, a shift value of zero should be used for a restrained structure and one for a free-free structure.

(ii) If the modes are needed in an intermediate range, a shift midway between the lowest and the highest expected eigenvalues should be used.

2.2.2 Mass and Stiffness Matrices - The iterative Rayleigh-Ritz analysis subroutines require as input mass and stiffness matrices of a structure. To reduce engineering time required to perform an analysis, subroutines were included to calculate mass and stiffness matrices for general standard structural elements. The basic idea behind the subroutines for mass and stiffness is outlined here for continuity.

Mass and stiffness matrices of the complete structure (fluid and bladder) are calculated using finite-element approach. In this approach, a continuous structure (fluid and bladder each separately) is assumed to be composed of simple, small structural elements - the so called finite elements - such as tetrahedron, pentahedron, triangular plates, quadrilateral plates, etc. The procedure to obtain the finite-element mass and stiffness matrices is based on kinetic and strain energy principles, respectively.

The kinetic energy for a complete structure may be expressed as

$$T = \frac{1}{2} \iiint \rho(x,y,z) \cdot \dot{\delta}^2(x,y,z,t)\, dx\, dy\, dz \tag{66}$$

where: T = kinetic energy

ρ = mass density

$\dot{\delta}$ = time rate change of deflection

t = time

x,y,z = global coordinates

The difficulty in integrating equation (66) is expressing the deflection $\delta\,(x,y,z,t)$ as a continuous function over the complete structure. In the finite-element approach, however, this apparent difficulty is circumvented by idealizing the structure to be comprised of many small structural elements for which $\delta\,(x,y,z,t)$ can be expressed as a continuous function. Thus, the expression (66) is valid for each of the finite-element of the structure. Then the kinetic energy of the structure is the summation of the kinetic energies of each of the finite elements, that is,

$$T = \sum T_i \tag{67}$$

where i refers to one particular finite element "i".
The common junction of finite elements is denoted as panel points, nodes or joints. Now, however, the deflection $\delta\,(x,y,z,t)$ is easily expressed as a simple function of the joint deflections. These element joint deflections are then generalized coordinates or degrees of freedom of the complete structure.

The approach is to derive the mass matrix for finite-element, "i", in a convenient local coordinate system and then transform it to the Global coordinate system. The technique is outlined here:

$$T_i = \frac{1}{2} \{\dot{h}_L(t)\}_i^T [m_L]_i \{\dot{h}_L(t)\}_i \tag{68}$$

where $[m_L]_i$ = the mass matrix in the local coordinate system for the ith element. This mass matrix is obtained by integration using an assumed displacement function. The discussion is deferred till later.

$\{\dot{h}_L(t)\}_i$ = the time rate of change of the joint deflections of finite-element, "i". This is in local system.

The deflections in the local coordinate systems are related to deflections in the global coordinate directions by a transformation matrix, $[\Upsilon]_i$. Thus

$$\{h_L(t)\}_i = [\Upsilon]_i \{h_G(t)\}_i \tag{69}$$

where $\{h_G(t)\}_i$ = the joint deflections of finite element, "i", in the global coordinate system.

Using equation (69) in equation (68)

$$T_i = \frac{1}{2}\{\dot{h}_G(t)\}_i^T [m_G]_i \{\dot{h}_G(t)\}_i \tag{70}$$

where $[m_G]_i = [\Upsilon]_i^T [m_L]_i [\Upsilon]_i$ (71)

is the mass matrix with respect to the global coordinate system for the ith finite-element. Further, all the elemental mass matrices are finally assembled to give the mass matrix of the total structure, as shown in equation (67).

The development of the finite-element stiffness matrices is similar to that of the mass matrices. The strain energy for the structure may be expressed as the summation of the strain energies of each finite elements. That is,

$$U = \Sigma U_i \tag{72}$$

As was done for the finite-element mass matrix, the stiffness matrix for finite-element, "i", is derived in a convenient local coordinate system. Thus,

$$U_i = \frac{1}{2}\{h_L(t)\}_i^T [K_L]_i \{h_L(t)\}_i \tag{73}$$

where $[k_L]_i$ = the stiffness matrix with respect to local coordinate directions for finite element, "i". This stiffness matrix is obtained by integration using an assumed displacement function. This will be discussed later.

$\{h_L(t)\}_i$ = the joint deflections of finite element, i, measured in local coordinate system.

The same transformation matrix, $[\Upsilon]_i$, which was used in equation (70) is used here to relate the deflections in local coordinates to deflections in global coordinates. Substitute then,

$$U_i = 1/2 \{h_G(t)\}_i^T [k_G]_i \{h_G(t)\}_i \tag{74}$$

where

$$[k_G]_i = [\Upsilon]_i^T [k_L]_i [\Upsilon]_i \tag{75}$$

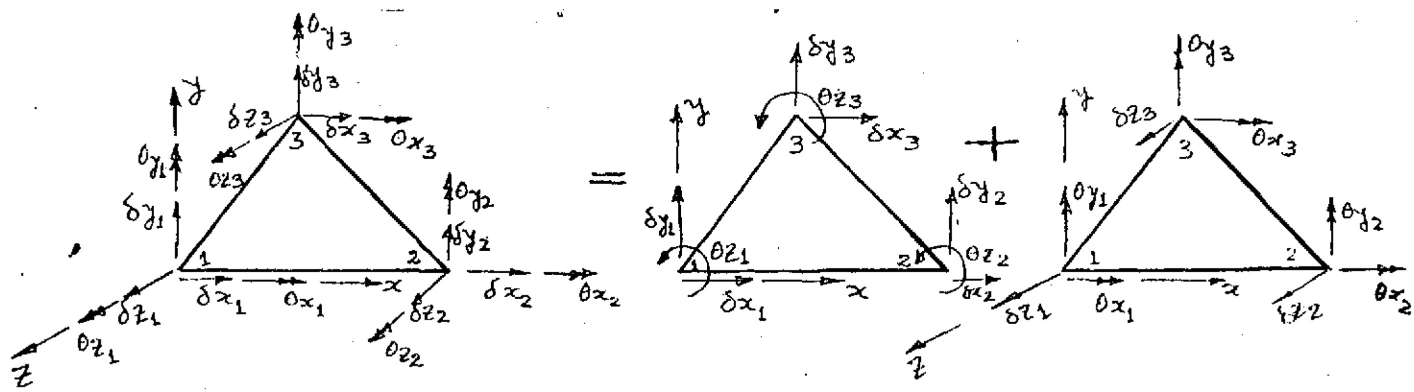
is the stiffness matrix with respect to the global coordinate system for the i<u>th</u> element.

Euler angle rotations at some joints (where the body coordinate is needed to be different than that of the global coordinates) are input in the program to allow the joint degree of freedom at these points to be different than that of global x,y,z directions.

However, in case of fluid, there is another item to be taken care of as far as stiffness matrix is concerned. This has to do with the surface elements of the fluid. The item concerned is known as gravitational potential. The energy contributed is known as the gravitational potential. This is caused by fluid movement in the gravitational field. Development of the gravitational potential stiffness effect is given in Section 2.2.4.

2.2.3 Stiffness and Mass Matrices in Local Coordinates (Solid Element)

Mass Matrix (Triangular Element)



Triangle Plate Element

Membrane Triangle Plate Element

Bending Triangle Plate Element

Figure - 2

The consistent mass matrix for a triangle plate element are calculated as a combination of the two following elements.

(i) consistent mass matrix for a membrane triangle plate element,

(ii) consistent mass matrix for a bending triangle plate element.

The elements of the mass matrix for the membrane triangle plate element represent the distributed mass properties of the triangle. These matrix elements are calculated by assuming a quadratic displacement field[3]. The elements of the mass matrix for the bending triangle plate element represent the distributed mass properties of the triangle. These matrix elements are calculated by assuming a cubic displacement field[4].

Stiffness Matrix: (triangular element)

The stiffness matrix for the triangle plate element is calculated in the same manner as the mass matrix and the technique and displacement fields are exactly the same.

Mass Matrix (quadrilateral element)

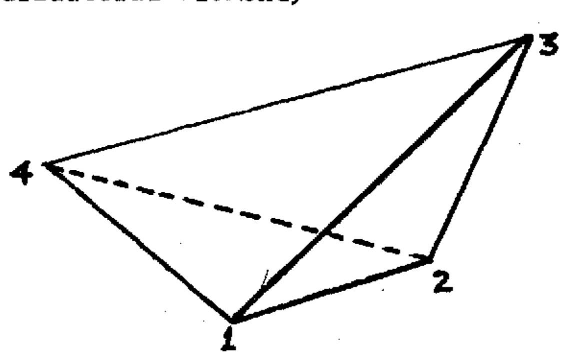


Figure 3

This mass matrix is calculated by taking the average of the four overlapping triangles created by the diagonals (1.3) and (2.4). The triangles are handled as discussed before. Thus, the quadrilateral case is nothing but a combination of triangular case.

Stiffness Matrix (quadrilateral element)

This is handled in exactly same fashion as the mass matrix case.

2.2.4 Stiffness and Mass Matrices in Local Coordinates (fluid finite element)

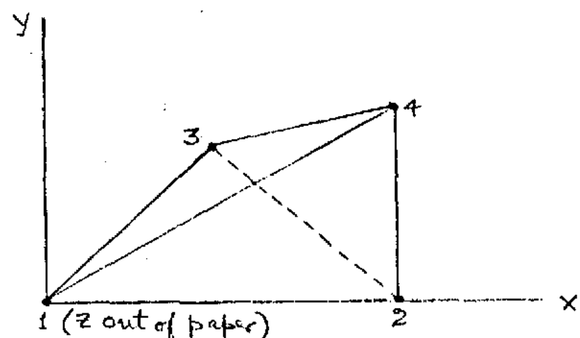


Figure - 4
Local Coordinate System for Tetrahedron Element

The basic fluid element is a tetrahedron; pentahedron elements and hexahedron elements are synthesized, simply by placing six and ten overlapping tetrahedrons together respectively and averaging the result. The averaging is carried out to eliminate the bias, if any.

For each tetrahedron element, a local cartesian coordinate system is defined so that vertex 1 is the origin, the x-axis includes vertex 2, vertex 3 lies in the x-y plane and vertex 4 always has a positive Z-coordinate (figure above).

This element considers a linear displacement field (constant strain). This is boundary conformable. The displacement field throughout the element is expressed in terms of coordinate locations and appears as

$$\bar{\omega}(x,y,z,t) = \bar{a}_0 + \bar{a}_1 x + \bar{a}_2 y + \bar{a}_3 z \tag{76}$$

The coefficients $\bar{a}_k(t)$, K = 0, 1, 2, 3 are eliminated in terms of the 12 vertex displacements.

The mass matrix for the fluid elements is obtained by expressing the kinetic energy

$$T = 1/2 \int_{Vol} \dot{\bar{\omega}} \cdot \dot{\bar{\omega}} \, \rho \, dv \tag{77}$$

where: T = kinetic energy

$\dot{\bar{\omega}} = \dot{\bar{a}}_0 + \dot{\bar{a}}_1 x + \dot{\bar{a}}_2 y + \dot{\bar{a}}_3 z$

ρ = mass density

This gives rise to a (12x12) mass matrix.

The stiffness matrix for the fluid element is obtained by expressing volumetric dilatation strain energy and gravitational potential energy in terms of vertex displacement coordinates

$$U_D = 1/2 \int_{Vol} k\theta^2 dv \tag{78}$$

$$U_g = 1/2 \rho g \int_{Area} (\bar{\omega} \cdot \bar{n}) (\bar{\omega} \cdot \bar{e}) \, ds \tag{79}$$

where U_D = volumetric dilatation energy

U_g = gravitational potential energy

k = fluid bulk modulus

θ = volumetric strain

$\bar{n}$ = unit outer normal

$\bar{e}$ = a unit vector parallel with the gravity vector $\bar{g}$, but of opposite sense, i.e., $\bar{e} = -\bar{g}/g$

An observation is made with respect to the gravitational potential energy as expressed in (79). Since it is a boundary conformable element, the surface integrals such as (79) will all cancel each other throughout the interior of the fluid in a container, since $\bar{n}$ on common element boundaries is equal and opposite. Thus, the gravitational potential energy will depend only on displacement coordinates at the boundary of the entire volume of the fluid, the free surface, the wetted container wall and also, in this case, the bladder. Also, for a rigid tank, $\bar{\omega} \cdot \bar{n}$ is non-zero only at the free surface where $\bar{e} = \bar{n}$; thus,

$$U_g = \frac{1}{2} \rho g \int_{\text{Free surface}} (\bar{\omega} \cdot \bar{n}) \, ds \tag{80}$$

Stiffness coefficients corresponding to gravitational potential and volumetric strain energies are thus derived.

2.3 Mechanical Equivalent - The mechanical equivalent in this case has been calculated on the following basis. The forces and moments developed by the model due to an external disturbance should correspond to the forces and moments exerted by the fluid under similar conditions.

Since any arbitrary liquid motion can be thought of as superposition of different slosh modes it suggests itself that the mechanical equivalent must consist of a series of spring mass systems, the masses corresponding to the effective amounts of liquid oscillating in different slosh modes. The frequencies of the mechanical equivalent must correspond to the frequencies of the elastomer for the mode they represent. However, to account for the part of the fluid that does not take part in the motion, one may also add a mass without a spring.

A short definition of the mechanical equivalent is as follows:
The equations of motion for a base driven elastic system are

$$\{\ddot{\xi}\} + [2\varsigma\omega]\{\dot{\xi}\} + [\omega^2]\{\xi\} = -[\Phi]^T[M][T]\{\ddot{q}_6\} \tag{81}$$

where:

ξ : modal coordinates,
$\dot{\xi}$: modal velocities,
$\ddot{\xi}$: modal accelerations,
Φ : discrete mode shapes,
T : rigid body transformation matrix,
$\ddot{q}_6$: discrete base accelerations (6 degrees of freedom),
ς : modal damping,
ω : frequencies,
M : mass matrix

The base reactions to support a given modal accelerations are

$$\{F\}_{Base} = [T]^T[M]\{\phi\}\ddot{\xi} \tag{82}$$

Noting that

$$\ddot{\xi} \propto \{\phi\}^T[M][T]\{\ddot{q}_6\} \tag{83}$$

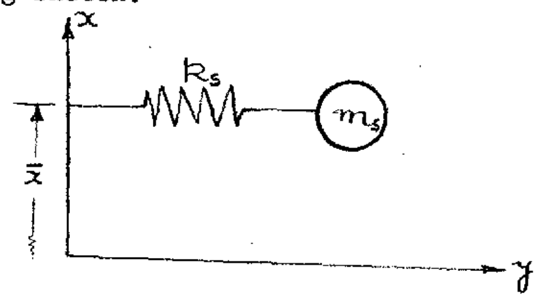
Substituting from(83) in(82)

$$\{F\}_{Base} \propto [T]^T[M]\{\phi\}\{\phi\}^T[M][T]\{\ddot{q}_6\} \tag{84}$$

where the proportionality parameter is a function of frequency.

The product

$[T]^T[M]\{\phi\}\{\phi\}^T[M][T]$ is a (6x6) matrix, in general, for a given mode. This may be called a matrix of forces and moments. The equivalent slosh parameters can be interpreted as shown in the following sketch.



where k_s: effective stiffness

m_s: effective slosh mass

$\bar{x}$: distance of line of action from the reference point.

The reference point is the same as the reference for the rigid body transformation.

This equivalent model is only valid for loads at the reference point. This may not be used for getting any detailed information concerning what is happening inside the elastic system.

In this particular case, the final product is a matrix (3x3). The term (2,2) is the slosh mass in the y-direction and the ratio of terms (2,3) and (2,2) is the distance $\bar{x}$. It should be noted that $\bar{x}$ may be outside the tank boundary. The slosh mass and $\bar{x}$ are printed out in the computer run. The slosh stiffness is the frequency squared because the modal displacements are normalized such that the generalized mass is unity. To obtain the mechanical equivalent slosh parameters of the preceding sketch, only y-translation and θ_z rotation need be considered in the rigid body transformation matrix T . Thus, the general 6x6 slosh matrix is a 2x2 matrix for the particular case considered here. The 1,1 term is the slosh mass in the lateral y-direction and the ratio of terms (1,2) and (1.1) is the distance $\bar{x}$. It should be noted that $\bar{x}$ may be outside the tank boundary. The slosh stiffness is the slosh frequency squared time the slosh mass, $\omega_s^2\ m_s$.

3. MODEL

The modeling of the system is achieved in two distinct parts. First, the static free surface of the fluid/bladder system, within the constraints, is established (refer to 2.1). This static free surface is subsequently used in modeling the total system as an assemblage of finite-elements for vibration.

3.1 <u>Finite Element Model of the System</u> - This is an axisymmetric system. The system thus can be subdivided into four quadrants, 1, 2, 3 and 4, as shown in the figure. Quadrants 2, 3 and 4 are reflected on to quadrant 1 with the following technique whereby the analysis of only one quadrant is necessary for the dynamic analysis of the total system. This renders the problem relatively small as far as sizes of the matrices are concerned.

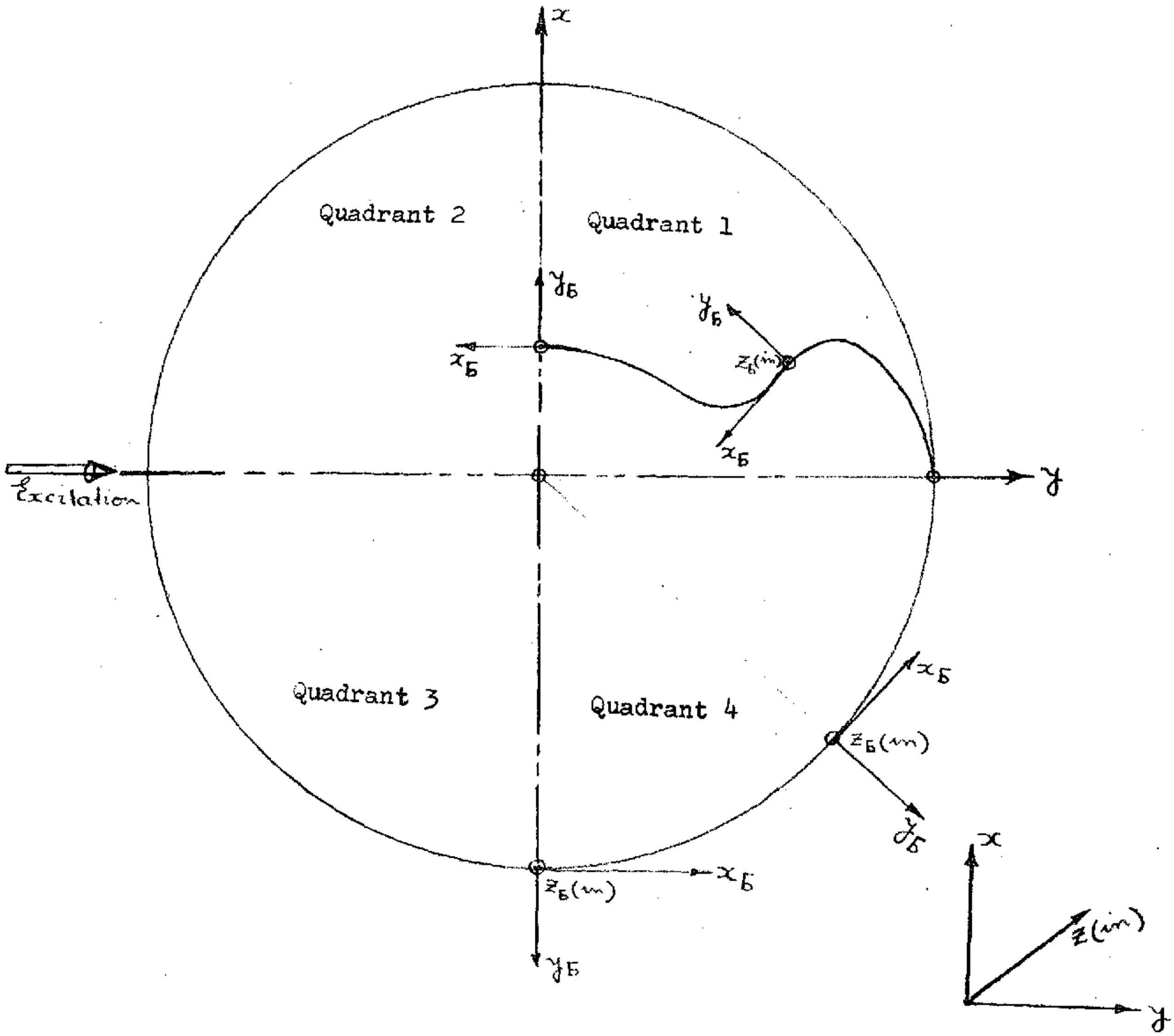


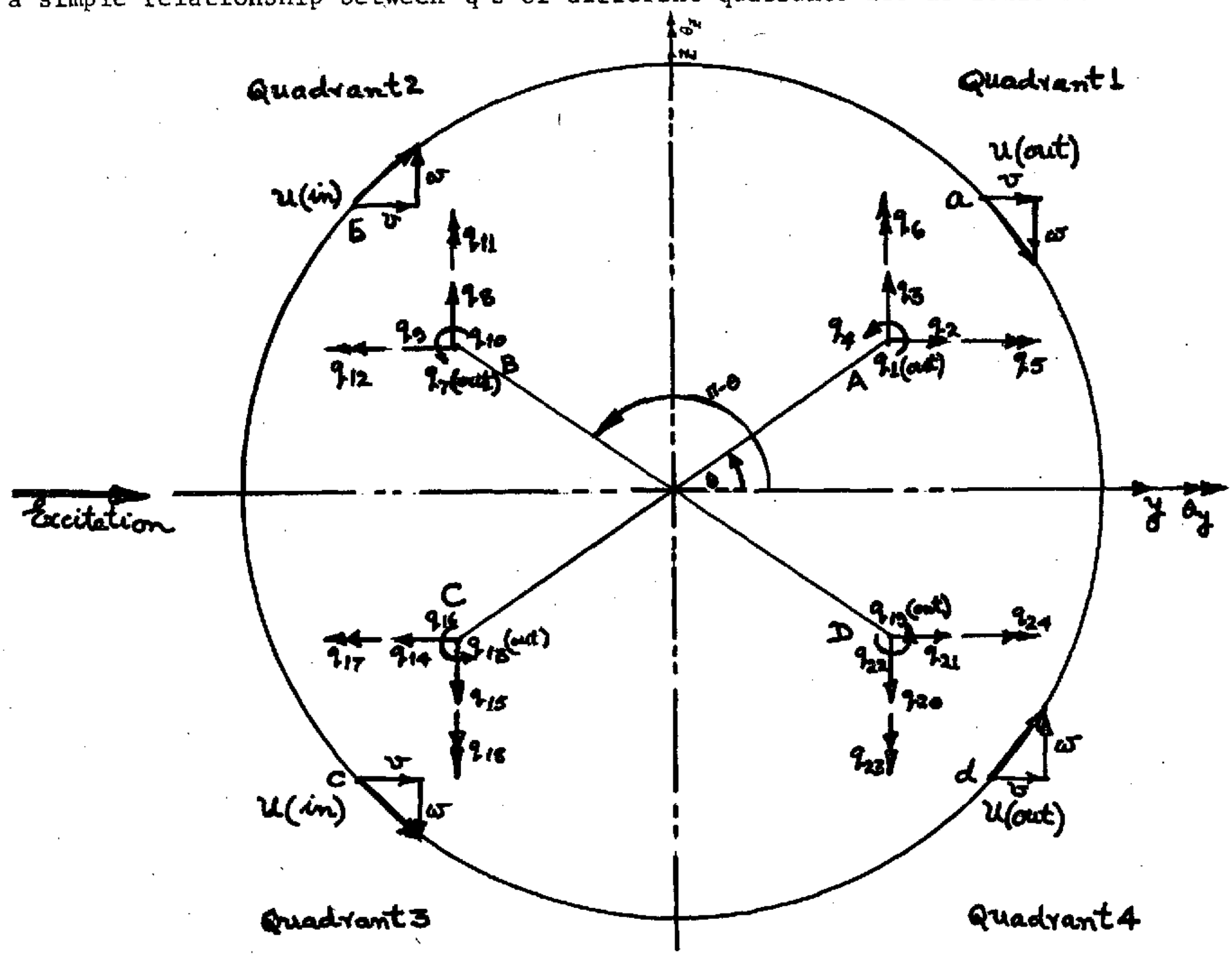
Figure 5. Quadrants and Coordinates Systems

x y z is the global coordinate system (right handed)

$x_Б\ y_Б\ z_Б$ is the so called body coordinate system (right handed)

The global and body coordinate system systems are t o different coordinate systems related through the Euler rotations. The global system remains fixed but the body coordinate system varies for different points. The global system is used for interior fluid and the body coordinate system is used on the boundary. The use of body coordinate system on the boundary makes the handling of the boundary condition easier. The body coordinates are shown in the figure for a few important points. The reflections of quadrants and the derivation of the Euler angles are discussed in the following pages.

3.2 Reflections of Quadrants 2, 3 and 4 on Quadrant 1 for Lateral Slosh - For the case of lateral slosh and the excitation direction a simple relationship between q's of different quadrants are as follows.



Lateral Slosh (Interior Points)

Figure 6

where: u, v and w are the velocities in the global x, y and z directions.

and a: a typical point in quadrant 1 (on the boundary)

b: a typical point in quadrant 2 (on the boundary)

c: a typical point in quadrant 3 (on the boundary)

d: a typical point in quadrant 4 (on the boundary)

q's: local coordinates

A: a typical point in quadrant 1 (interior)

B: a typical point in quadrant 2 (interior)

C: a typical point in quadrant 3 (interior)

D: a typical point in quadrant 4 (interior)

$$\left\{\begin{matrix} q_7 \\ q_8 \\ q_9 \\ q_{10} \\ q_{11} \\ q_{12} \end{matrix}\right\}_2 = \left[\begin{array}{|c|c|c|c|c|c|} -1 & & & & & \\ & & -1 & & & \\ & -1 & & & & \\ & & & 1 & & \\ & & & & & 1 \\ & & & & 1 & \end{array}\right] \left\{\begin{matrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{matrix}\right\}_1 \qquad (85)$$

$$\left\{\begin{matrix} q_{13} \\ q_{14} \\ q_{15} \\ q_{16} \\ q_{17} \\ q_{18} \end{matrix}\right\}_3 = \left[\begin{array}{|c|c|c|c|c|c|} -1 & & & & & \\ & -1 & & & & \\ & & -1 & & & \\ & & & -1 & & \\ & & & & -1 & \\ & & & & & -1 \end{array}\right] \left\{\begin{matrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{matrix}\right\}_1 \qquad (86)$$

$$\left\{\begin{matrix} q_{19} \\ q_{20} \\ q_{21} \\ q_{22} \\ q_{23} \\ q_{24} \end{matrix}\right\} = \left[\begin{array}{|c|c|c|c|c|c|} 1 & & & & & \\ & & 1 & & & \\ & 1 & & & & \\ & & & -1 & & \\ & & & & & -1 \\ & & & & -1 & \end{array}\right] \left\{\begin{matrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{matrix}\right\} \qquad (87)$$

Thus, it is apparent

$$\{q\}_3 = -\{q\}_1 \qquad (88)$$

$$\{q\}_4 = -\{q\}_2 \qquad (89)$$

There exists a relation between q's and global coordinates. They are (writing them in terms of their velocity or displacement components)

$$\begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{Bmatrix}_1 = \begin{Bmatrix} u \\ v \\ w \\ \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix} \tag{90}$$

$$\begin{Bmatrix} q_7 \\ q_8 \\ q_9 \\ q_{10} \\ q_{11} \\ q_{12} \end{Bmatrix}_2 = \begin{Bmatrix} u \\ w \\ -v \\ \theta_x \\ \theta_z \\ -\theta_y \end{Bmatrix} \tag{91}$$

Recognizing the relations (88) and (89), the other quadrants relations are not followed through any further. From relations (85) and (91) the relation between the degrees of freedom of quadrants 1 and 2 for interior of the fluid can be written as

$$\begin{Bmatrix} u \\ v \\ w \\ \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix}_2 = \begin{bmatrix} -1 & & & & & \\ & 1 & & & & \\ & & -1 & & & \\ & & & 1 & & \\ & & & & -1 & \\ & & & & & 1 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix}_1 \tag{92}$$

Thus

$$\{u\}_2 = [T_{21}]\{u\}_1 \tag{93}$$

where

$$\{u\} = \begin{Bmatrix} u \\ v \\ w \\ \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix}$$

$$[T_{21}] = \begin{bmatrix} -1 & & & & & \\ & 1 & & & & \\ & & -1 & & & \\ & & & 1 & & \\ & & & & -1 & \\ & & & & & 1 \end{bmatrix}$$

If the boundary degrees of freedom were also in the global comparison system, the same transformation could be applied to them. But as the situation stands, they are in the body coordinates system which in general will have a different transformation than $[T_{21}]$.

4. RESULTS

The results of the study has been obtained in three distinct steps. They are: (1) Static Equilibrium Shape, (2) Vibration Analysis and, (3) Equivalent Slosh Mass for Dynamic Characteristics of the System. The static equilibrium shapes obtained in the first part are fed into the second part for establishing the grid pattern for finite-element analysis of the system.

4.1 Static Equilibrium Shapes - The technique used for this has been discussed in Section 2.1. A computer program has been developed for this purpose which is listed in Section 7.

The static equilibrium shapes for fill conditions of 80%, 60%, 50%, 40% and 20% for 1g case have been obtained using the available computer program. However, for the case of 10^{-5}g conditions, only fill ratios of 80% and 60% have been successfully run through the program. The other cases of 50%, 40% and 20% were extrapolated from available results. This was achieved with the help of graphical representation between the percentage fill and the coefficients of the power series, assumed for the static surface, see Figures 11G through 14G. The graphical representations of the static free surface and the coefficients are presented in Figures 1G through 14G.

The values of other parameters used in all of these are:

(i) Radius of the sphere = 8 inches (20.32 cm)

(ii) Young's modulus for the bladder material = 200 lb/in^2 (1.378951×10^6 N/m^2)

(iii) Thickness of the bladder = 0.06 inches (0.1524 cm)

(iv) Mass density for mercury = 0.0013 lb/in/sec^2 (0.00059 Kgm)

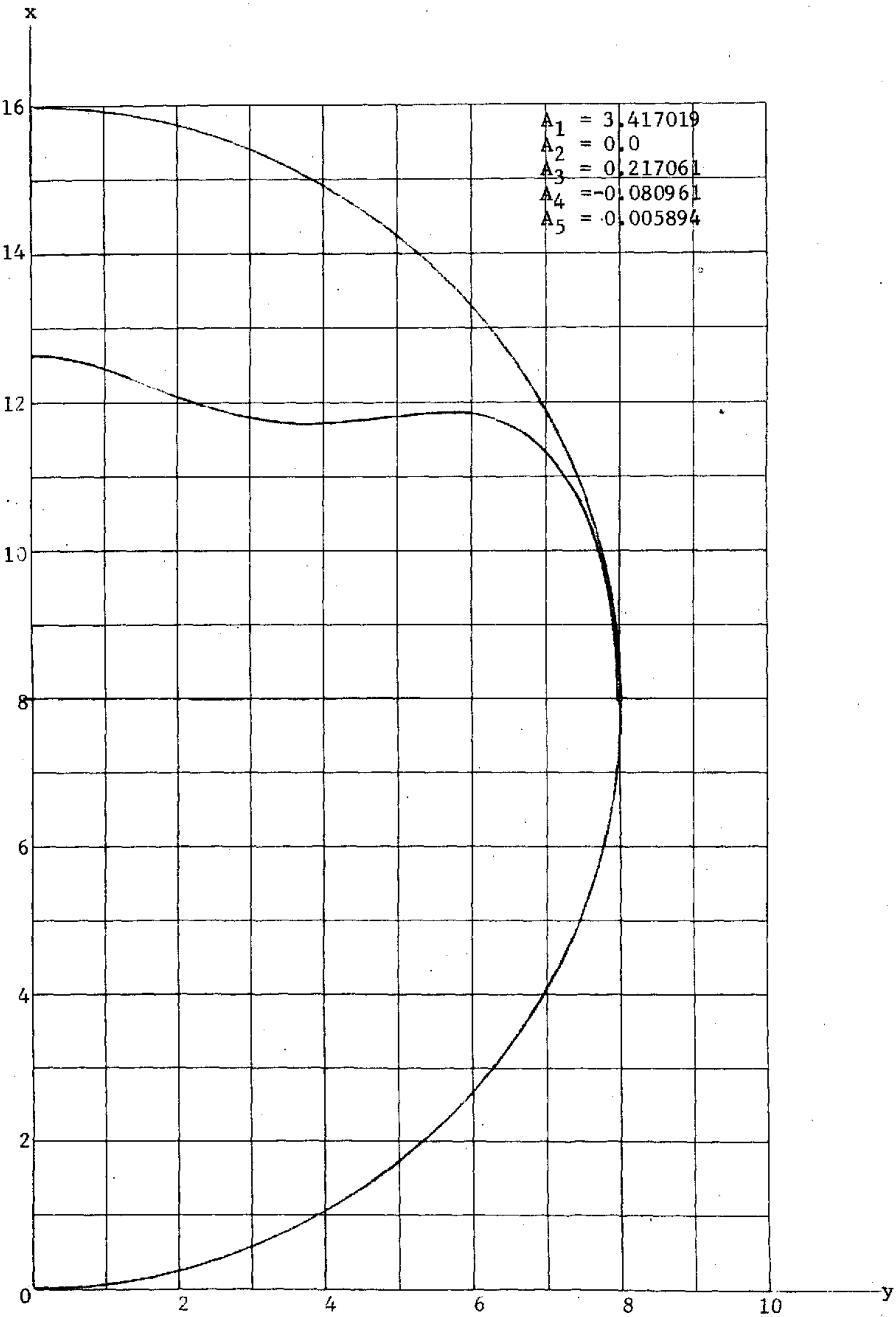
(v) Ullage pressure = 0.10 lb/in^2 (689.48 N/m^2)

and the assumptions made are:

(vi) Bladder attachment level = diametral plane

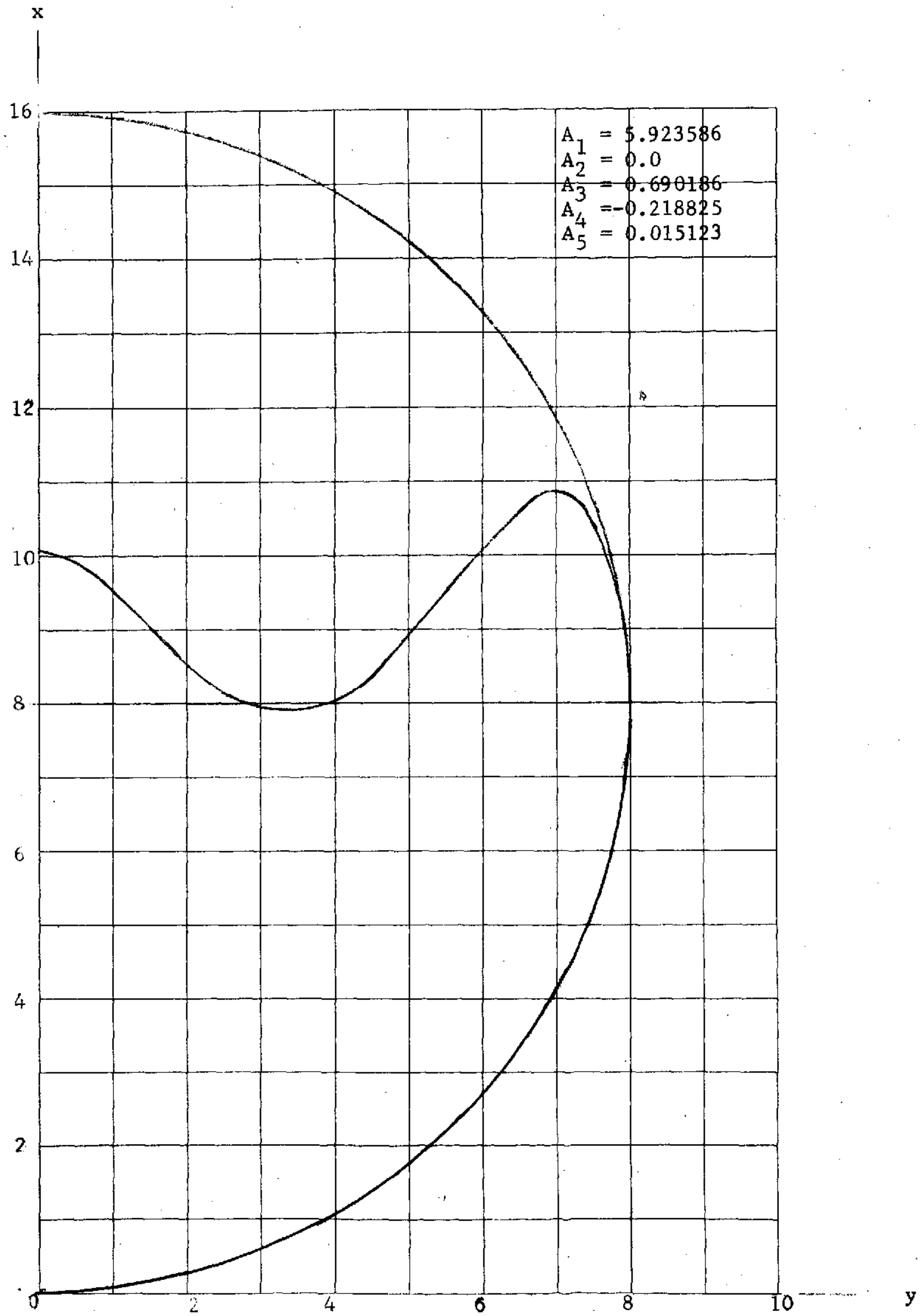
(vii) Order of the polynomial = 4

Figures 11G through 14G may be used to establish the coefficients which define the static free surface for any gravity and any percentage fill without making any more computer runs.



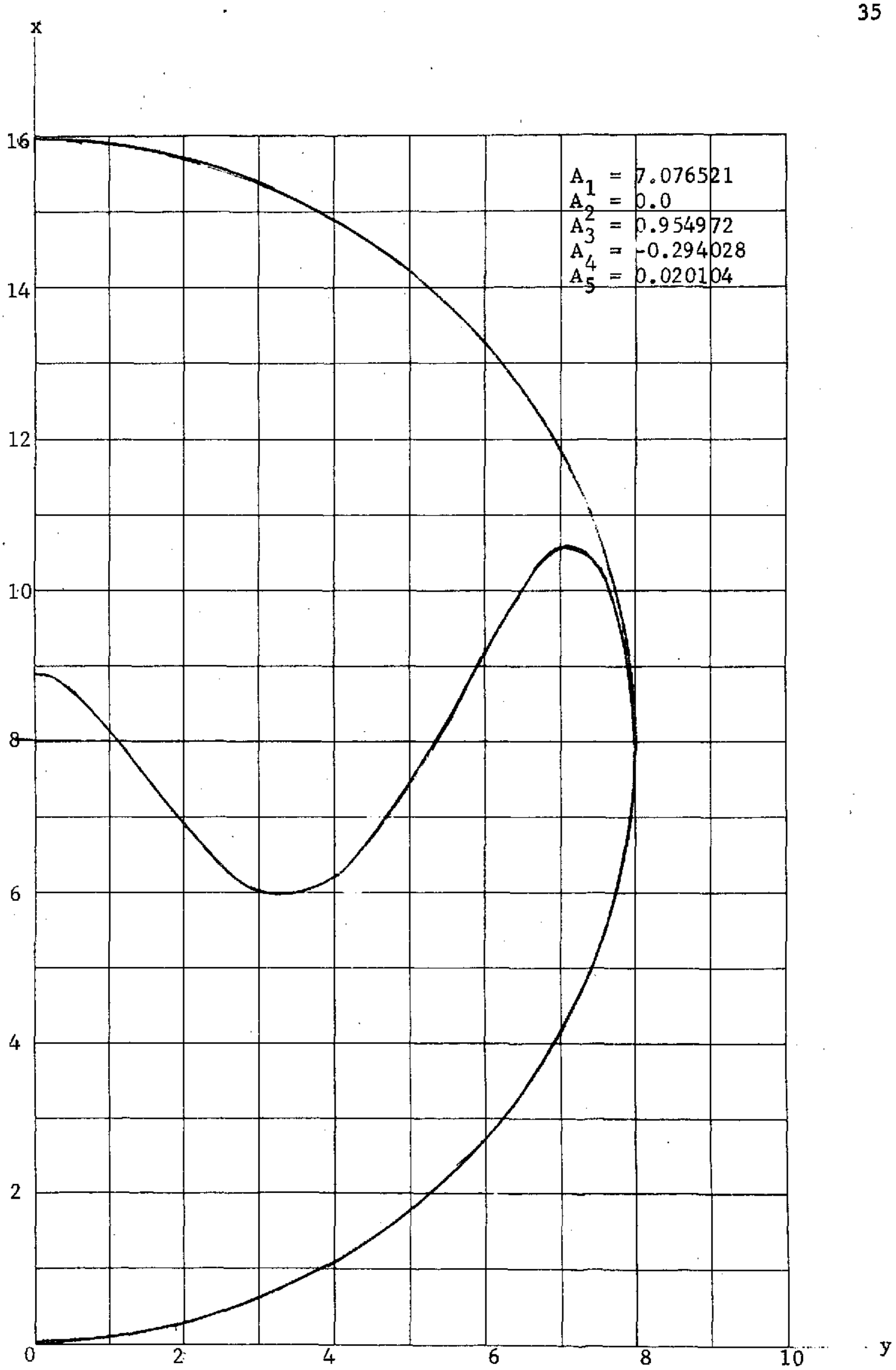
STATIC FREE SURFACE FOR 80% FULL, 1g

FIGURE 1G



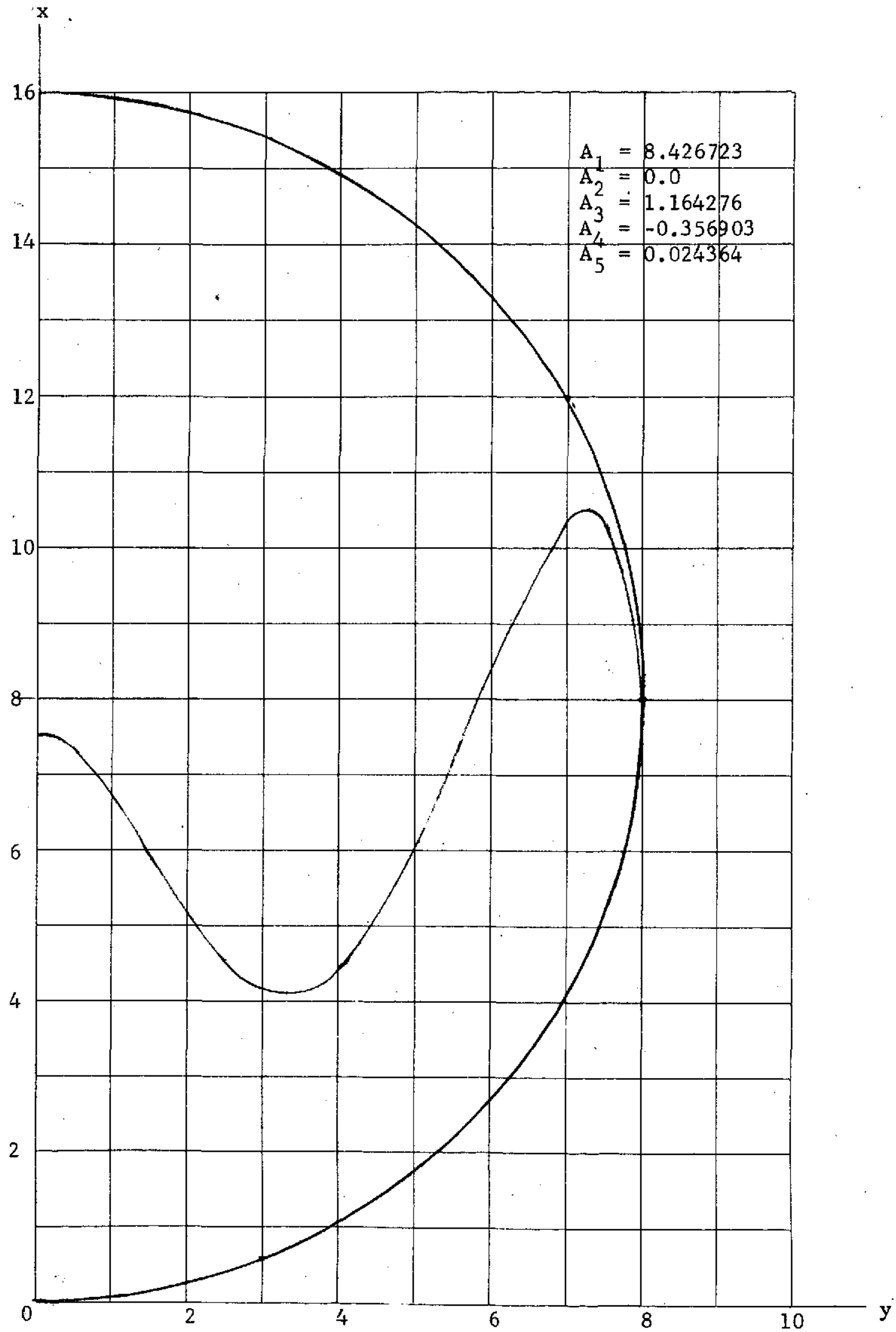
STATIC FREE SURFACE FOR 60% FULL, 1g

FIGURE 2G



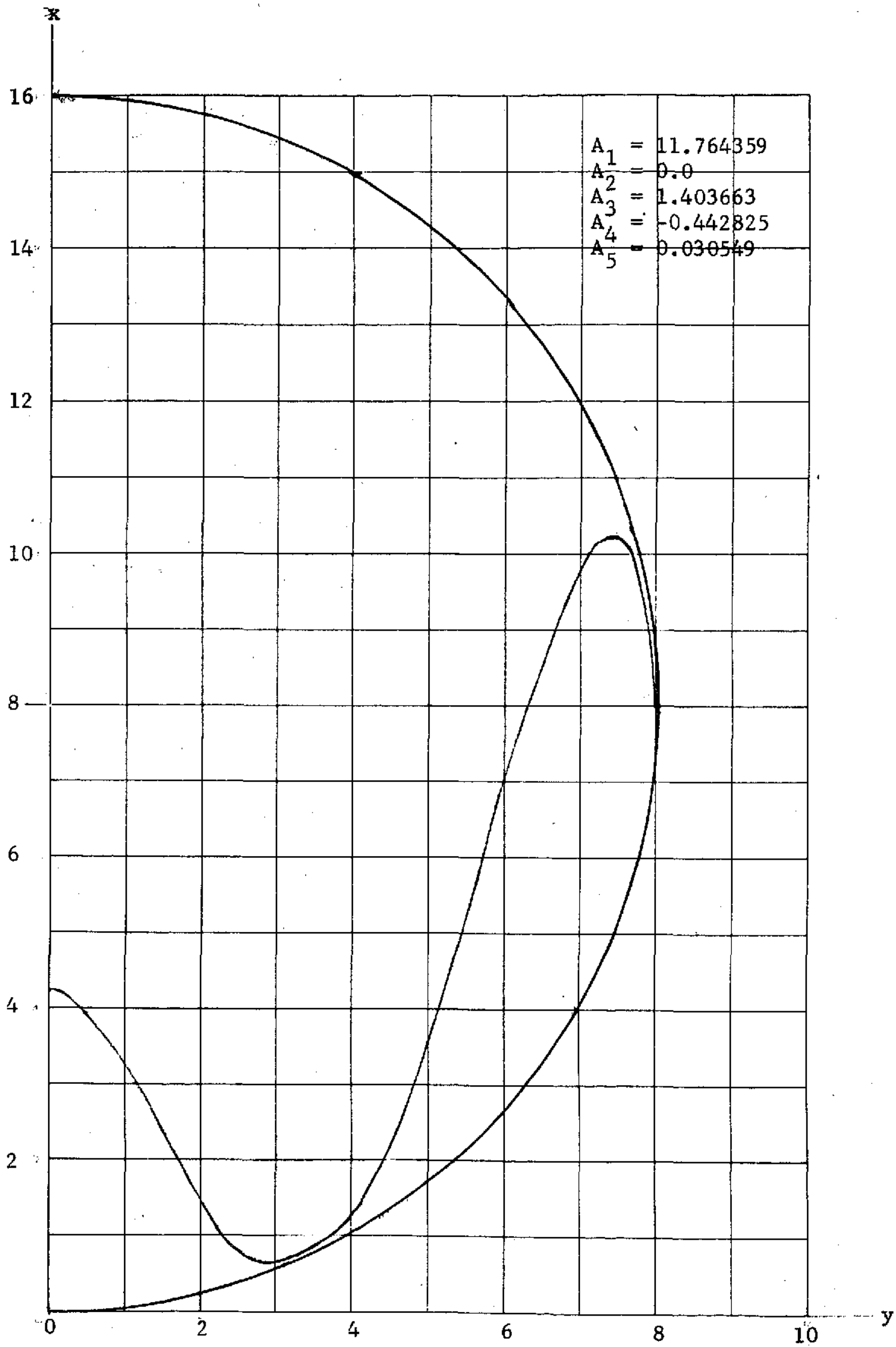
STATIC FREE SURFACE FOR 50% FULL, 1g

FIGURE 3G



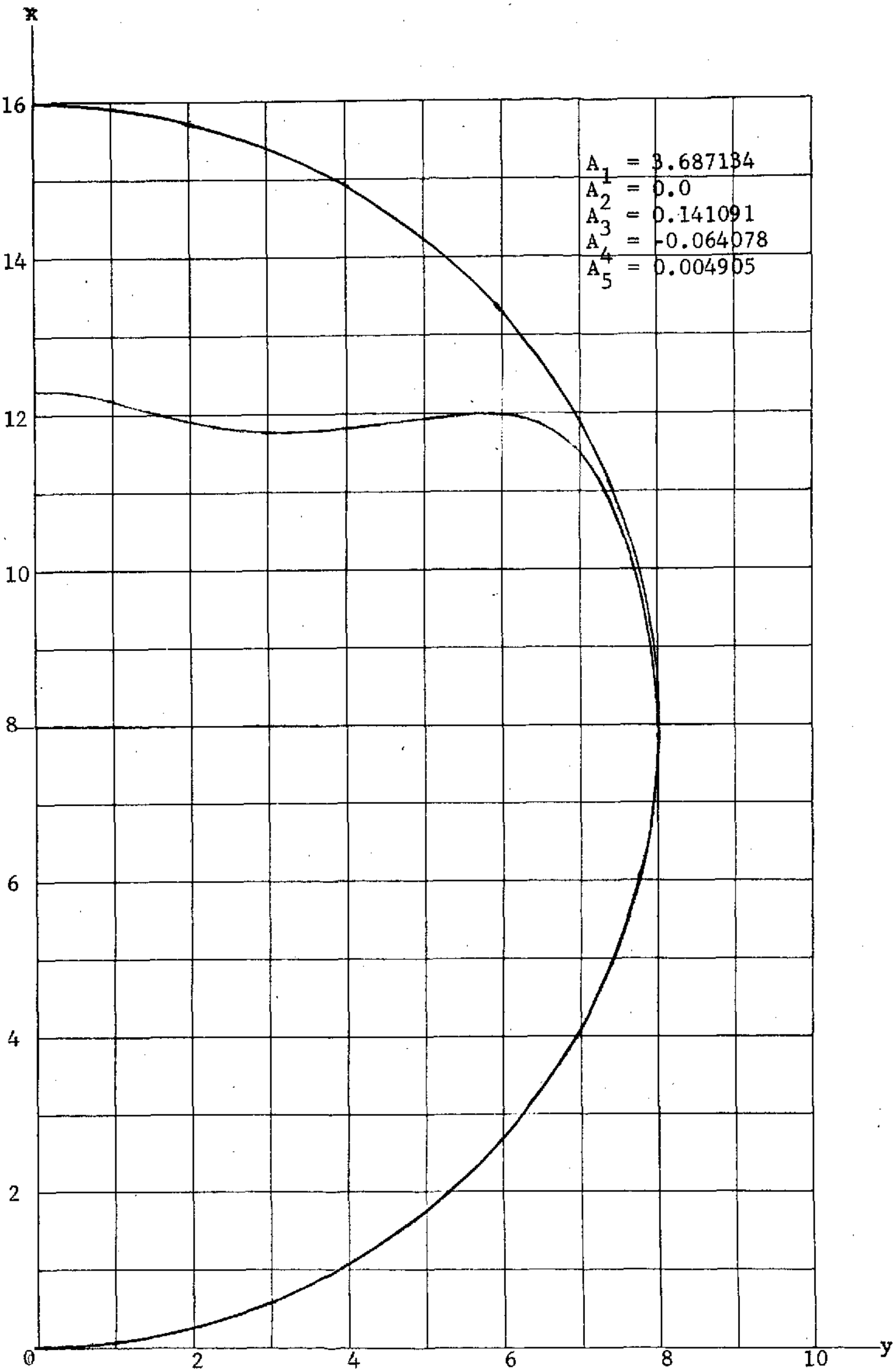
STATIC FREE SURFACE FOR 40% FULL, 1g

FIGURE 4G



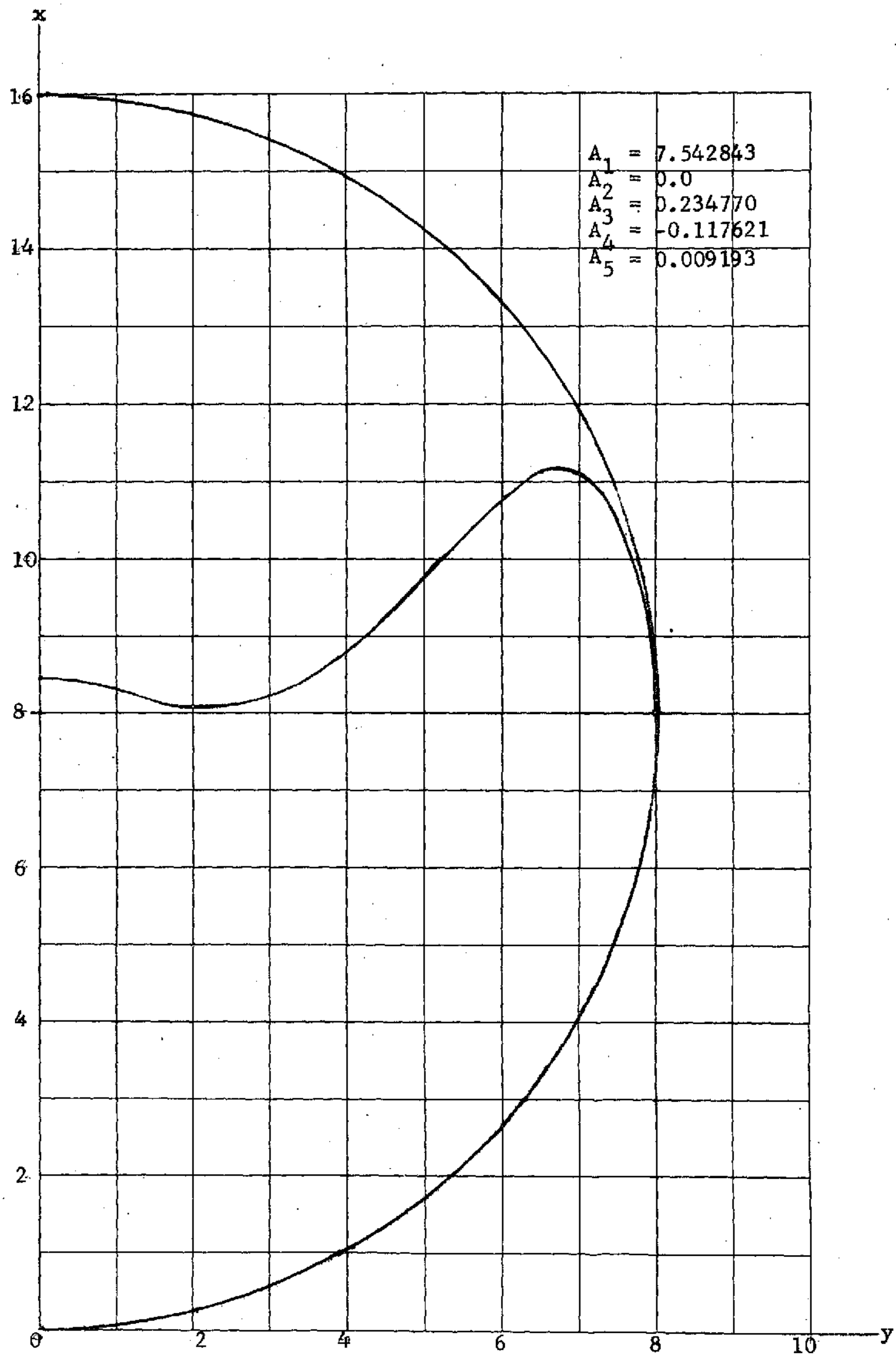
STATIC FREE SURFACE FOR 20% FULL, 1g

FIGURE 5G



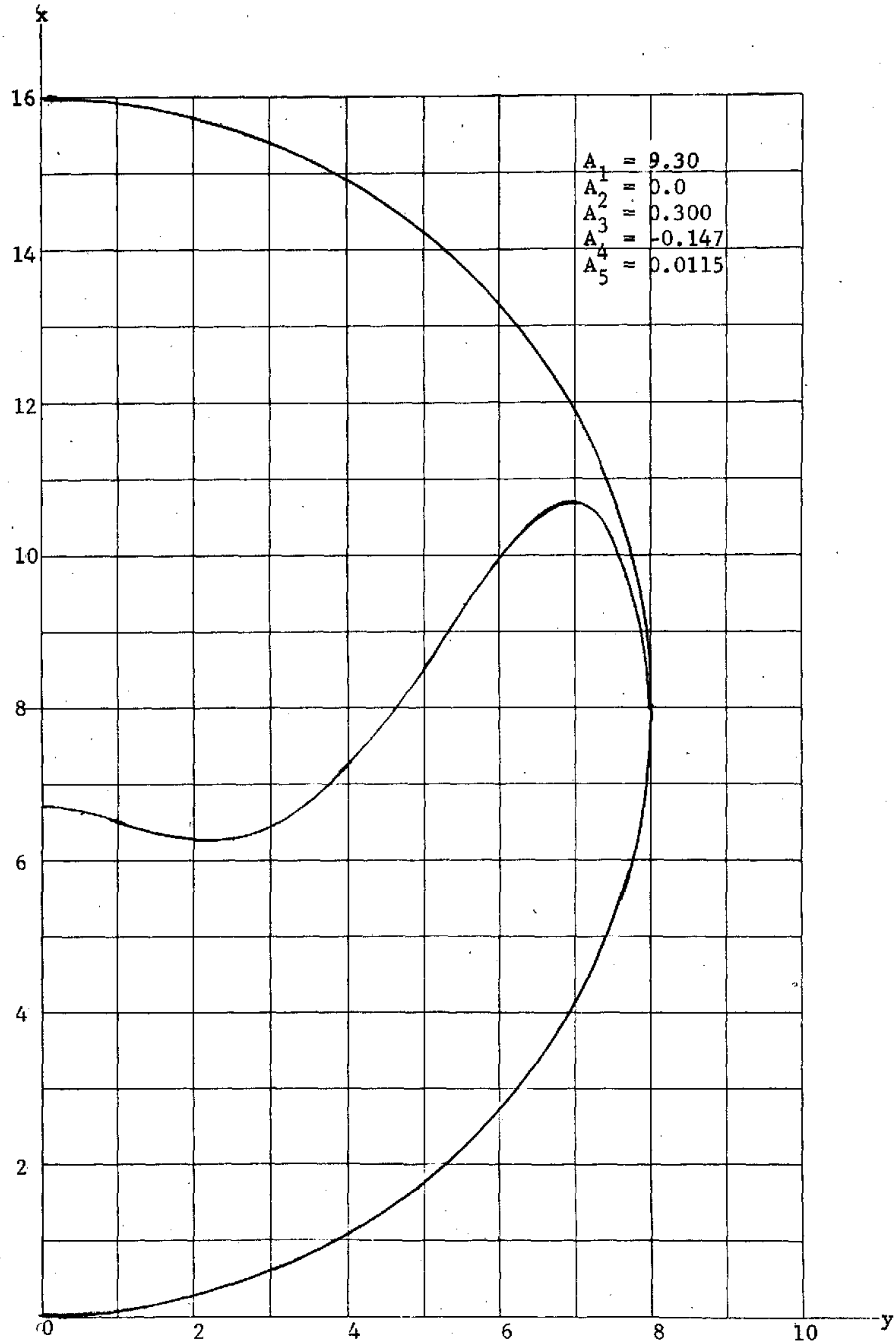
STATIC FREE SURFACE FOR 80% FULL, 10^{-5}g

FIGURE 6G



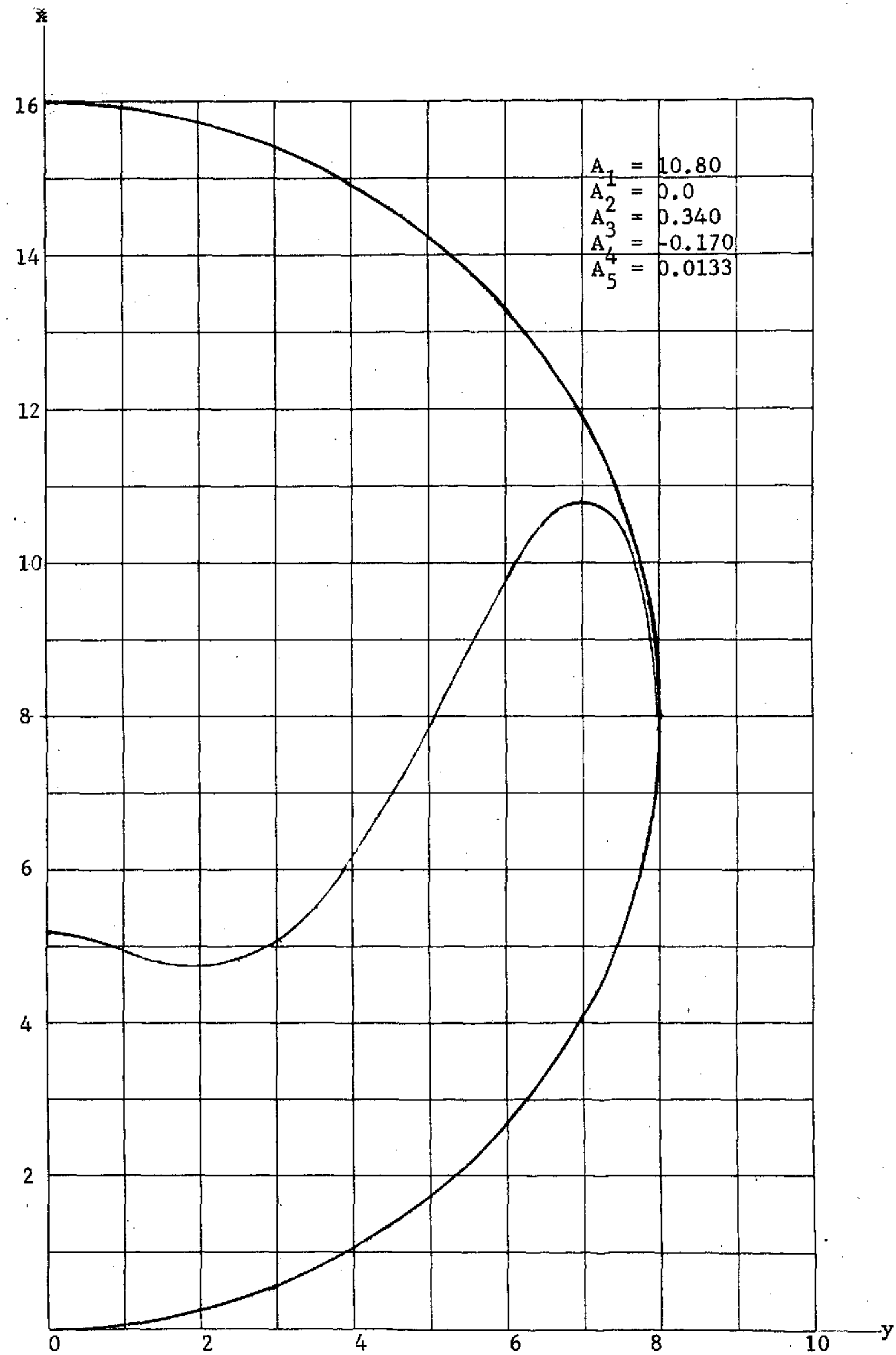
STATIC FREE SURFACE FOR 60% FULL, 10^{-5}g

FIGURE 7G



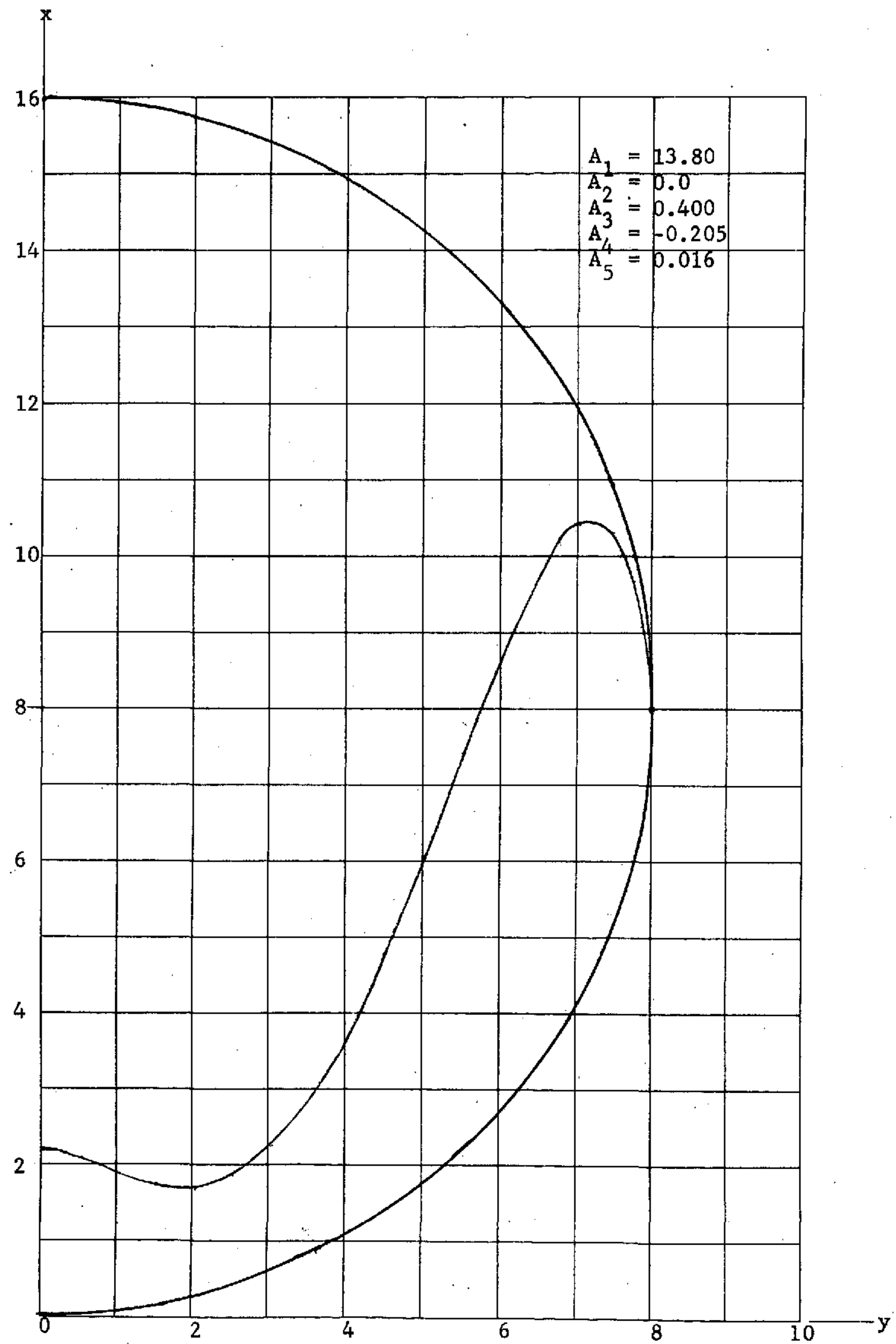
STATIC FREE SURFACE FOR 50% FULL, 10^{-5}g

FIGURE 8G



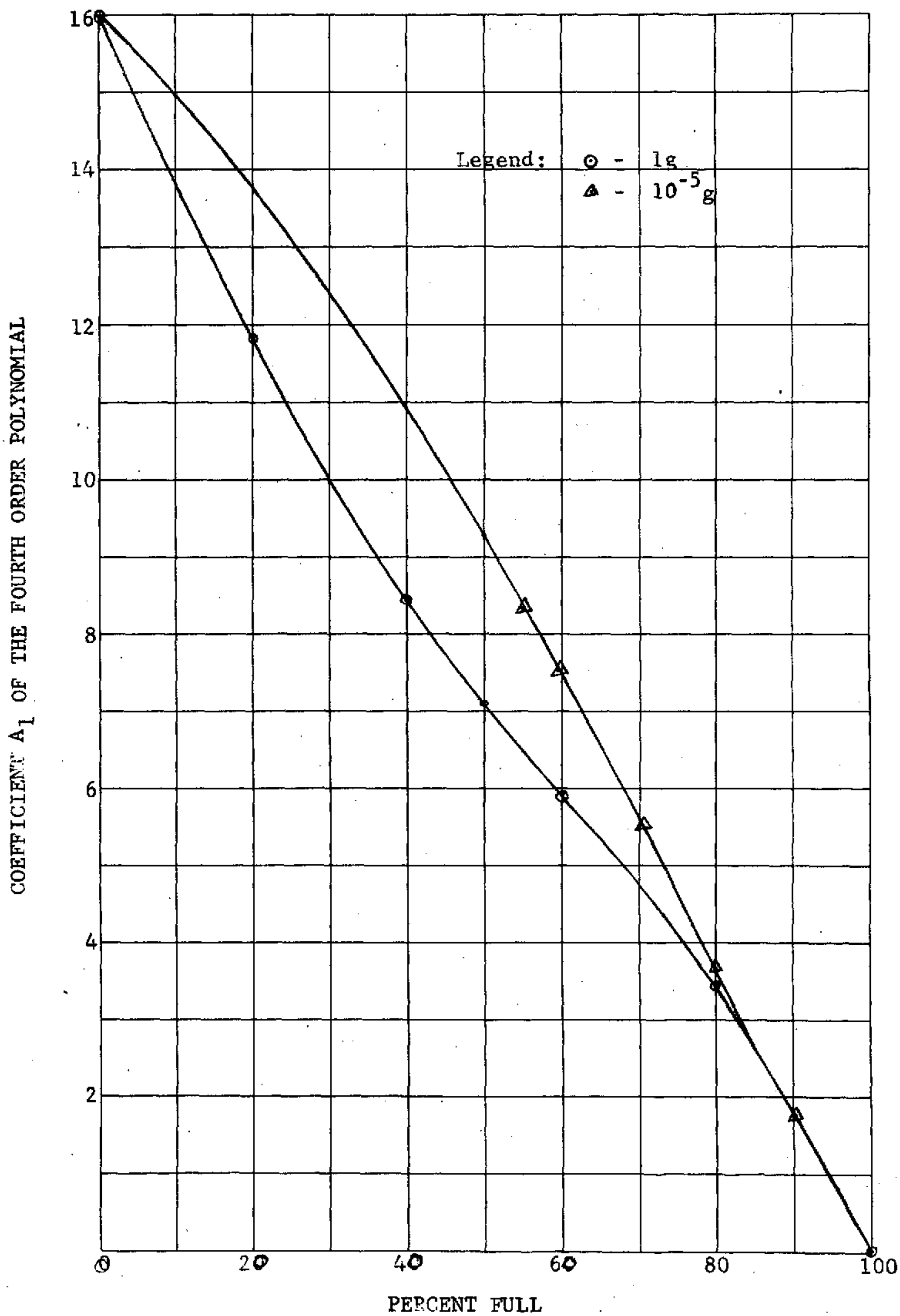
STATIC FREE SURFACE FOR 40% FULL, 10^{-5}g

FIGURE 9G



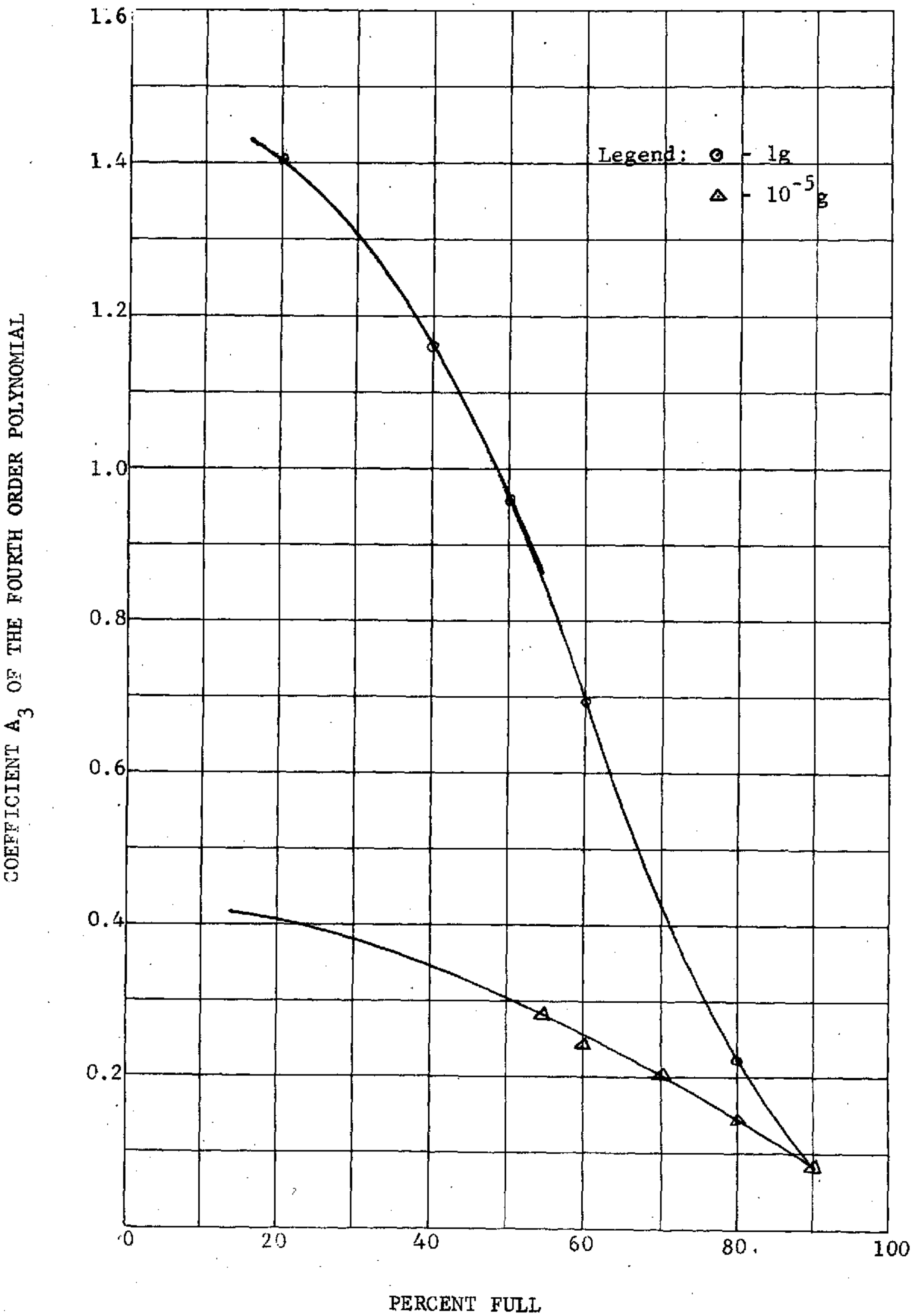
STATIC FREE SURFACE FOR 20% FULL, 10^{-5}g

FIGURE 10G



Legend: ⊙ - 1g
△ - 10^{-5}g
COEFFICIENT A_1 OF THE FOURTH ORDER POLYNOMIAL
16
14
12
10
8
6
4
2
0
20
40
60
80
100
PERCENT FULL

FIGURE 11G



COEFFICIENT A_3 OF THE FOURTH ORDER POLYNOMIAL
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0
20
40
60
80
100
Legend: ⊙ - 1g
△ - 10^{-5}g
PERCENT FULL

FIGURE 12G

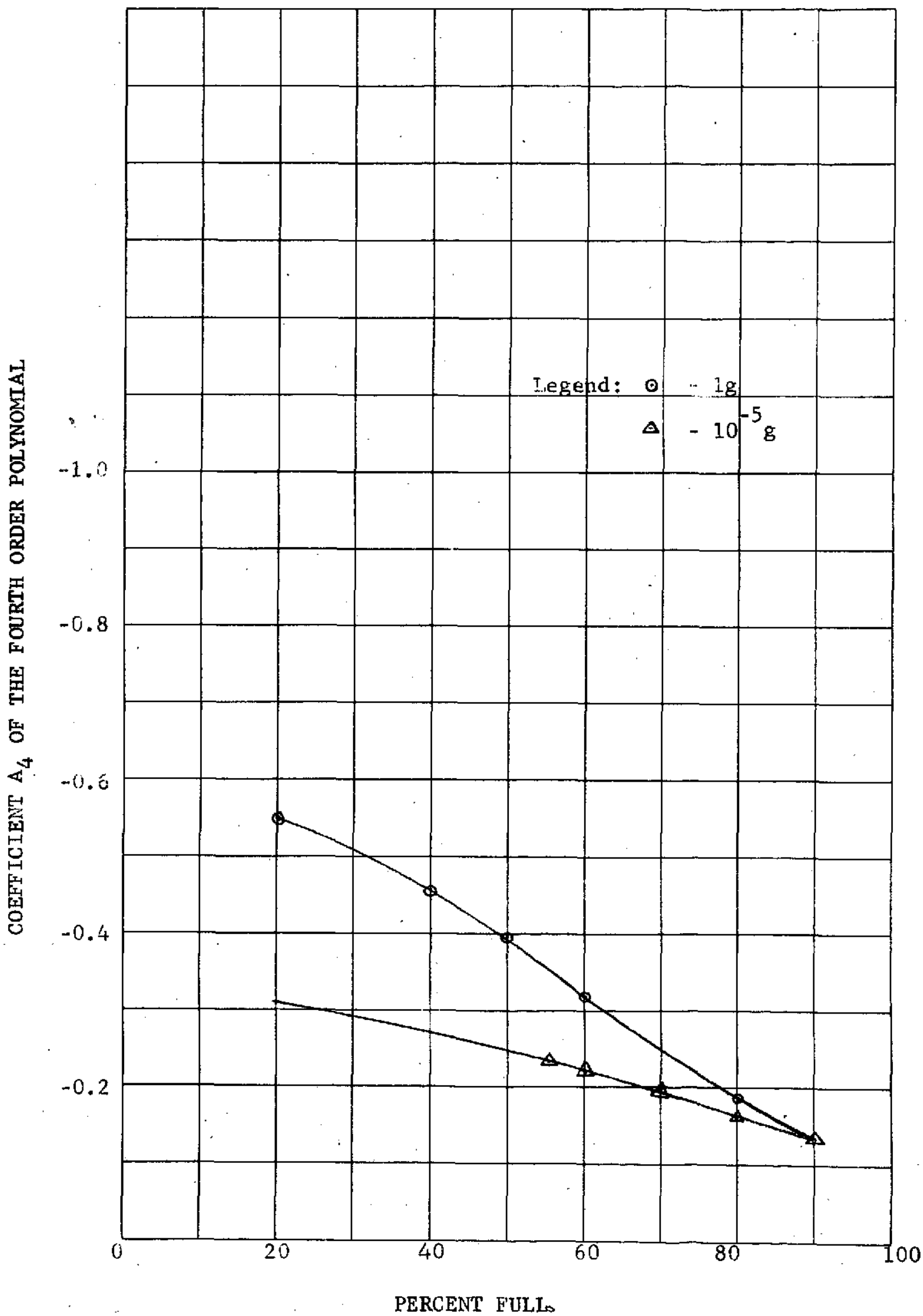

Legend: ⊙ - 1g
△ - 10^{-5}g
-1.0
-0.8
-0.6
-0.4
-0.2
0
20
40
60
80
100
COEFFICIENT A_4 OF THE FOURTH ORDER POLYNOMIAL
PERCENT FULL

FIGURE 13G

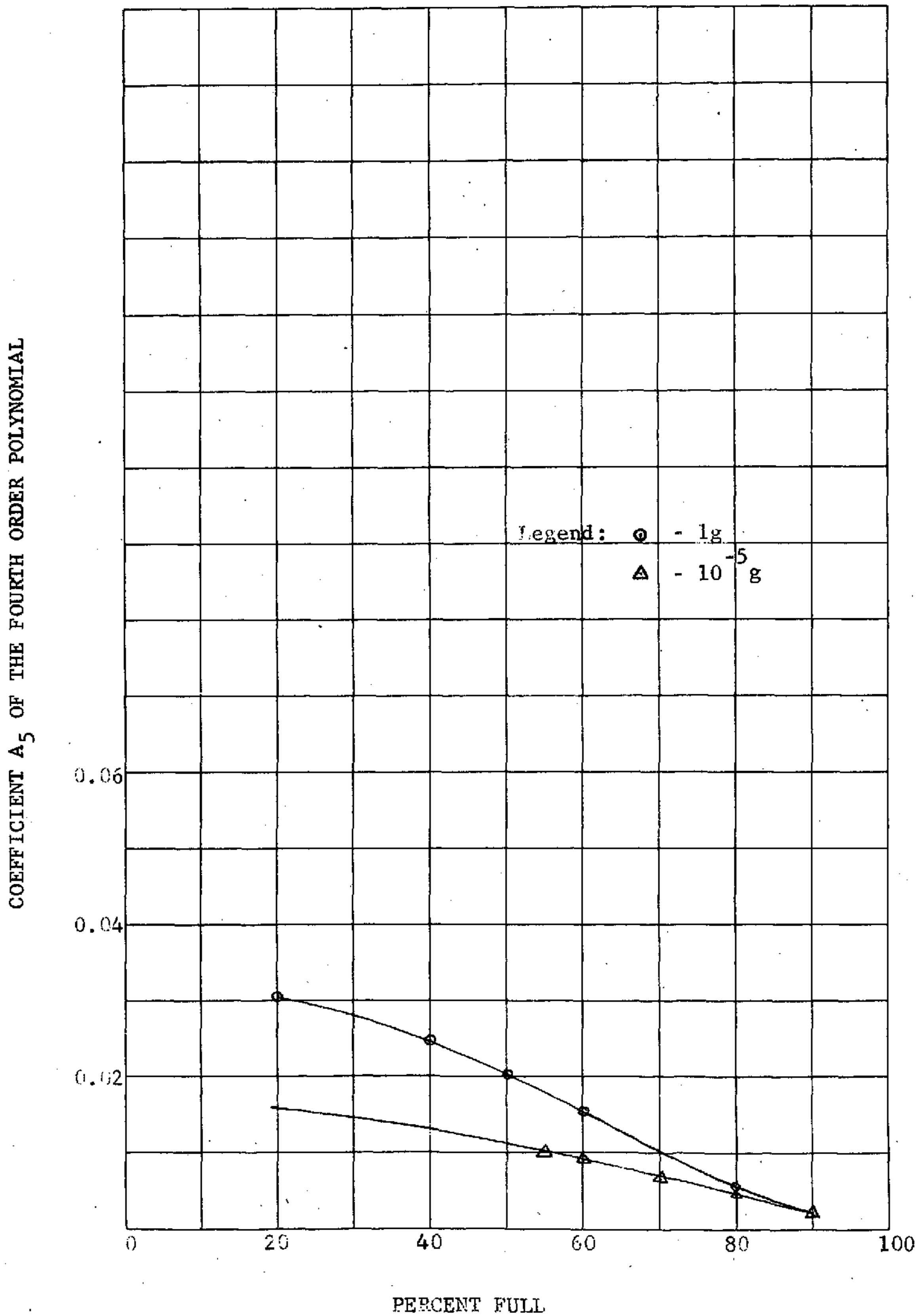

COEFFICIENT A_5 OF THE FOURTH ORDER POLYNOMIAL
0.06
0.04
0.02
0
20
40
60
80
100
Legend: ⊙ - 1g
△ - 10^{-5}g
PERCENT FULL

FIGURE 14G

4.2 Vibration Analysis - The iterative Rayleigh-Ritz technique used to calculate the vibration mode shapes and frequencies has been described in Section 2.2.1. The computer subroutine "YMODE2" used to calculate these modal values already exists at the MSFC computer center and is, thus, omitted from the program listings of Appendix A2.

The calculated vibration mode shapes were used in the calculation of the mechanical equivalent slosh parameters which are given in Section 4.3. The calculated vibration frequencies are listed in Section 4.3.

4.3 Mechanical Equivalent Slosh - The technique used to calculate the mechanical equivalent slosh parameters has been described in Section 2.3. The computer subroutine "MECHEQ" used to calculate these parameters is included in Appendix A2.

A great deal of difficulty was encountered in the identification of the slosh modes among the many vibration modes calculated. It was decided to identify as slosh modes those modes having the largest slosh mass. The following tables give the slosh frequencies, masses, stiffnesses and attach stations. It should be noted that numerical precision of the computer casts some doubt on the accuracy of these modal results.

Table 4.3-1

SLOSH PARAMETERS
80% FILL CONDITION
1g ACCELERATION

f (Hz)	Mass lb-sec²/in	Stiffness lb/in	Attach Station ($\bar{x}$) in
1.0756	.0037	.1690	4.2
1.0828	.0022	.1018	10.0
1.0946	.0002	.0104	6.8
1.1193	.0001	.0044	8.9
1.1235	.0014	.0698	5.0
1.1337	.0011	.0558	6.2
1.1709	.0008	.0411	1.5
1.2429	.0180	1.0978	6.0
1.2471	.0014	.0860	4.3
1.2640	.0250	1.5768	3.9
1.3188	.0130	.8926	5.7
1.3276	.0035	.2435	4.8
1.3709	.0120	.8903	5.1
1.4495	.0007	.0581	11.4
1.4625	.0003	.0262	8.4
1.4729	.0350	2.9976	3.7
1.4952	.0001	.0097	-4.0

Table 4.3-2

SLOSH PARAMETERS
60% FILL CONDITION
1g ACCELERATION

f (Hz)	Mass lb-sec²/in	Stiffness lb/in	Attach Station ($\bar{x}$) in
1.0805	.0008	.0350	22.7
1.1020	.0093	.4458	2.0
1.1284	.0660	3.3176	5.1
1.1384	.0002	.0112	7.3
1.1741	.0230	1.2516	7.7
1.1871	.0007	.0384	-3.8
1.2307	.0041	.2452	3.7
1.2355	.0550	3.3143	3.5
1.2412	.0240	1.4596	1.7
1.2647	.0830	5.2406	1.5
1.2748	.0056	.3592	-1.3
1.2764	.0420	2.7014	0.8
1.3093	.1400	9.4748	2.3
1.4149	.0000	.0001	-94.0

Table 4.3-3

SLOSH PARAMETERS
50% FILL CONDITION
1g ACCELERATION

f (Hz)	Mass lb-sec^2/in	Stiffness lb/in	Attach Station ($\bar{x}$) in
1.5270	.0003	.0285	-3.6
1.5692	.0062	.6027	10.2
1.5888	.0060	.5979	12.3
1.8572	.0002	.0245	-2.6
1.9030	.0180	2.5735	9.6
2.0489	.0160	2.6518	6.7
2.0725	.0000	.0100	4.7

Table 4.3-4

SLOSH PARAMETERS
40% FILL CONDITION
1g ACCELERATION

f (Hz)	Mass lb-sec^2/in	Stiffness lb/in	Attach Station ($\bar{x}$) in
1.0594	.0004	.0199	3.3
1.0988	.0001	.0028	-3.3
1.1035	.0210	1.0258	1.3
1.1071	.0008	.0397	5.0
1.1674	.0004	.0200	-2.8
1.2077	.0017	.0978	10.4
1.2184	.0001	.0068	35.6
1.2429	.0002	.0127	-0.3
1.2570	.0005	.0336	28.0
1.2705	.0001	.0072	25.0
A jump was made to higher frequencies			
5.4081	.0003	.3938	37.2
6.0773	.0000	.0032	11.4
7.0748	.0000	.0000	-177.0
8.1944	.0004	1.0516	28.3
8.2019	.0001	.2094	27.5

Table 4.3-5

SLOSH PARAMETERS
20% FILL CONDITION
1g ACCELERATION

f (Hz)	Mass lb-sec^2/in	Stiffness lb/in	Attach Station ($\bar{x}$) in
1.1170	.0019	.0936	3.7
1.1209	.0060	.2971	3.1
1.1250	.0017	.0084	-0.4
2.6893	.0099	2.8272	1.9
3.3176	.0000	.0162	-16.6

Table 4.3-6

SLOSH PARAMETERS
80% FILL CONDITION
10^{-5}g ACCELERATION

f (Hz)	Mass lb-sec^2/in	Stiffness lb/in	Attach Station ($\bar{x}$) in
0.9374	.0390	1.3469	3.7
0.9803	.0055	.2080	1.0
1.1229	.0006	.0319	17.0
1.1420	.0022	.1151	-2.8
1.1697	.0023	.1235	9.6
1.2470	.0000	.0011	22.0
1.2914	.0012	.0823	9.4
1.3485	.6025	43.2559	7.7
1.3604	.3385	24.7301	7.3
1.3964	.0113	.8699	9.2
1.3967	.0064	.4940	3.4
1.4706	.0011	.0981	2.2
1.5501	.0037	.3546	4.6
1.6660	.0003	.0285	-26.2

Table 4.3-7

SLOSH PARAMETERS
60% FILL CONDITION
10^{-5}g ACCELERATION

f (Hz)	Mass lb-sec^2/in	Stiffness lb/in	Attach Station ($\bar{x}$) in
0.9789	.0006	.0246	18.2
1.0190	.0033	.1334	5.4
1.0461	.0530	2.2743	6.4
1.1590	.0040	.2098	5.9
1.2204	.0001	.0072	8.1

Table 4.3-8

SLOSH PARAMETERS
50% FILL CONDITION
10^{-5}g ACCELERATION

f (Hz)	Mass lb-sec^2/in	Stiffness lb/in	Attach Station ($\bar{x}$) in
1.0629	.0008	.0380	-10.9
1.1177	.0180	.8636	-0.4
1.1392	.0043	.2197	-9.9
1.1759	.0095	.5195	0.7
1.1766	.0030	.1620	-2.8
1.2032	.0094	.5393	0.3
1.2247	.0000	.0002	17.9
1.2859	.0030	.1972	-2.4
1.3917	.0000	.0039	5.0
1.3942	.0020	.1569	-2.6
1.3949	.0027	.2103	3.8
1.4356	.0030	.2431	-2.0
1.5645	.0003	.0320	-10.6
1.5788	.0022	.2118	2.8
1.6007	.0001	.0066	-22.0

Table 4.3-9

SLOSH PARAMETERS
40% FILL CONDITION
10^{-5}g ACCELERATION

f (Hz)	Mass lb-sec^2/in	Stiffness lb/in	Attach Station ($\bar{x}$) in
1.0036	.0038	.1519	-6.3
1.0191	.0001	.0021	-39.0
1.0238	.0006	.0212	2.9
1.0384	.0026	.1099	-8.7
1.0660	.0015	.0685	1.4
1.0973	.0010	.0475	17.7
1.1022	.0036	.1745	13.3
1.1441	.0001	.0078	42.9
1.1496	.0004	.0200	12.1
1.1772	.0045	.2457	2.9
1.1823	.0220	1.2059	9.4
1.2267	.0000	.0001	10.3
1.2270	.0008	.0471	0.7
1.2514	.0250	1.5727	3.6
1.2555	.0027	.1698	20.1
1.2879	.0060	.3929	4.7
1.3154	.0008	.0548	-0.8
1.3403	.0001	.0070	-3.5
1.3754	.0004	.0306	-0.5

Table 4.3-10

SLOSH PARAMETERS
20% FILL CONDITION
10^{-5}g ACCELERATION

f (Hz)	Mass lb-sec/in	Stiffness lb/in	Attach Station ($\bar{x}$) in
1.0284	.0008	.0314	-4.9
1.0661	.0030	.1363	-2.6
1.1009	.0000	.0003	-2.07
1.1260	.0000	.0006	55.9
1.1440	.0012	.0615	2.0
1.1624	.0003	.0154	23.5
1.1703	.0017	.0939	1.1
1.1933	.0004	.0235	-9.7
1.2512	.0000	.0025	-48.5
1.3078	.0000	.0023	7.5
1.3175	.0052	.3592	-1.5
1.3381	.0019	.1320	5.2
1.3401	.0011	.0746	-10.1
1.3642	.0007	.0510	-29.8
1.4709	.0001	.0048	-67.8

5. CONCLUSIONS AND COMMENTS

It is felt that a significant contribution was made in this study to the state of the art in finite element fluid analysis. As with all new investigative analytical studies, review of the work performed reveals that although significant advances were made, several items that should have been studied further or have been performed in a different manner if time had permitted.

Several different approaches were developed and programmed to calculate the static equilibrium shape of the fluid/bladder system. Most of them did not perform satisfactorily due to the numerical or convergence problems. Finally, being limited by the time available, a two-dimensional representation - an infinite channel - was chosen for the representation of the equilibrium shape as the only approach showing a reasonably good convergence. A future effort should review this approach critically and perhaps extend it to a 3-dimensional representation.

A second item that should be investigated is the use of double precision in the computer programs, particularly for the calculation of vibration modal properties. This study pointed up the possible need for double precision because a bladder with small modulus of elasticity was used in the analysis. Thus, clear separation of the fluid slosh modes from fluid circulation modes was clouded. In addition, use of low shift values, λ_s , (see Section 2.2.1) sometimes resulted in failure to decompose the dynamical matrix $[K] - \lambda_s [M]$ due to singularity. This is obviously a computer accuracy problem.

The data generator computer subroutine, used to calculate joint X, Y, Z locations, degree of freedom values, Euler angles, and finite element joint numbers has some limitations which should be removed in future studies. One of these limitations is the requirement for vertical radial cuts. This was used to minimize user input but it became obvious later that odd shaped elements resulted. A more general data generator should be coded to allow more user control on the shape of the elements.

The representation of the lateral slosh by a spring mass system has been achieved on the assumption that the modes are completely uncoupled. A detailed investigation of slosh equivalent modeling techniques should be persued in this direction in a follow-on effort.

REFERENCES

1. "Modal Analysis of Structures by an Iterative Rayleigh-Ritz Technique", by J. R. Admire

2. "Advanced Engineering Mathematics", by C. R. Wylie, Jr., McGraw-Hill, New York, 1960

3. "Linear Strain Membrane Triangle Element", by W. A. Benfield and C. S. Bodley, MMC, DM 169

4. "Triangle Elements in Plate Elements", by C. P. Bazely, Y. K. Cheung, B. M. Irons, and O. C. Zienkiewicz, AFFDL-TR-66-80, November 1966

BIBLIOGRAPHY

1. "Sloshing Liquids in Axisymmetric Ellipsoidal Tanks," by Jasti V. Rattayya, AIAA paper no. 65-114, AIAA 2nd Aerospace Sciences Meeting, New York, N.Y./January 25-27, 1965.

2. "Analytical Representation of Lateral Sloshing by Equivalent Mechanical Models," by Franklin T. Dodge, The Dynamic Behavior of Liquids in Moving Containers, edited by H. Norman Abramson, NASA SP-106.

3. "Slosh Testing of a Spherical Mercury Propellant Tank With Positive-Expulsion Diaphragm," by R. G. Ross, Jr. and J. R. Womack, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, July 15, 1973.

4. "Experimentally Determined Pendulum Analogy of Liquid Sloshing in Spherical and Oblate-Spheroidal Tanks," by I. E. Summer, Lewis Research Center, Cleveland, Ohio. NASA TN D-2737.

5. "Low Gravity Liquid-Vapor Interface Shapes in Axisymmetric Containers and a Computer Solution," by L. J. Hastings and R. Rutherford, III, George C. Marshall Space Flight Center, Alabama, NASA TM X-53790.

6. "Liquid Oscillation Frequencies in Tilted Cylinders for Five Cross-Section Shapes," by S.Thornton and Frank Bugg, Aero-Astrodynamics Laboratory, October 1, 1971, NASA TM X-64618.

7. "An Evaluation of the Low G Propellant Behavior of a Space Vehicle During Waiting Orbit," by R. S. Ryan and H. Buchanan, Aero-Astrodynamics Laboratory, NASA TM X-53476.

8. "Finite Element Methods in Stress Analysis," edited by Ivar Holand and Kolbein Bell, published by TAPIR, the Technical University of Norway, Trondheim-Norway, 1972.

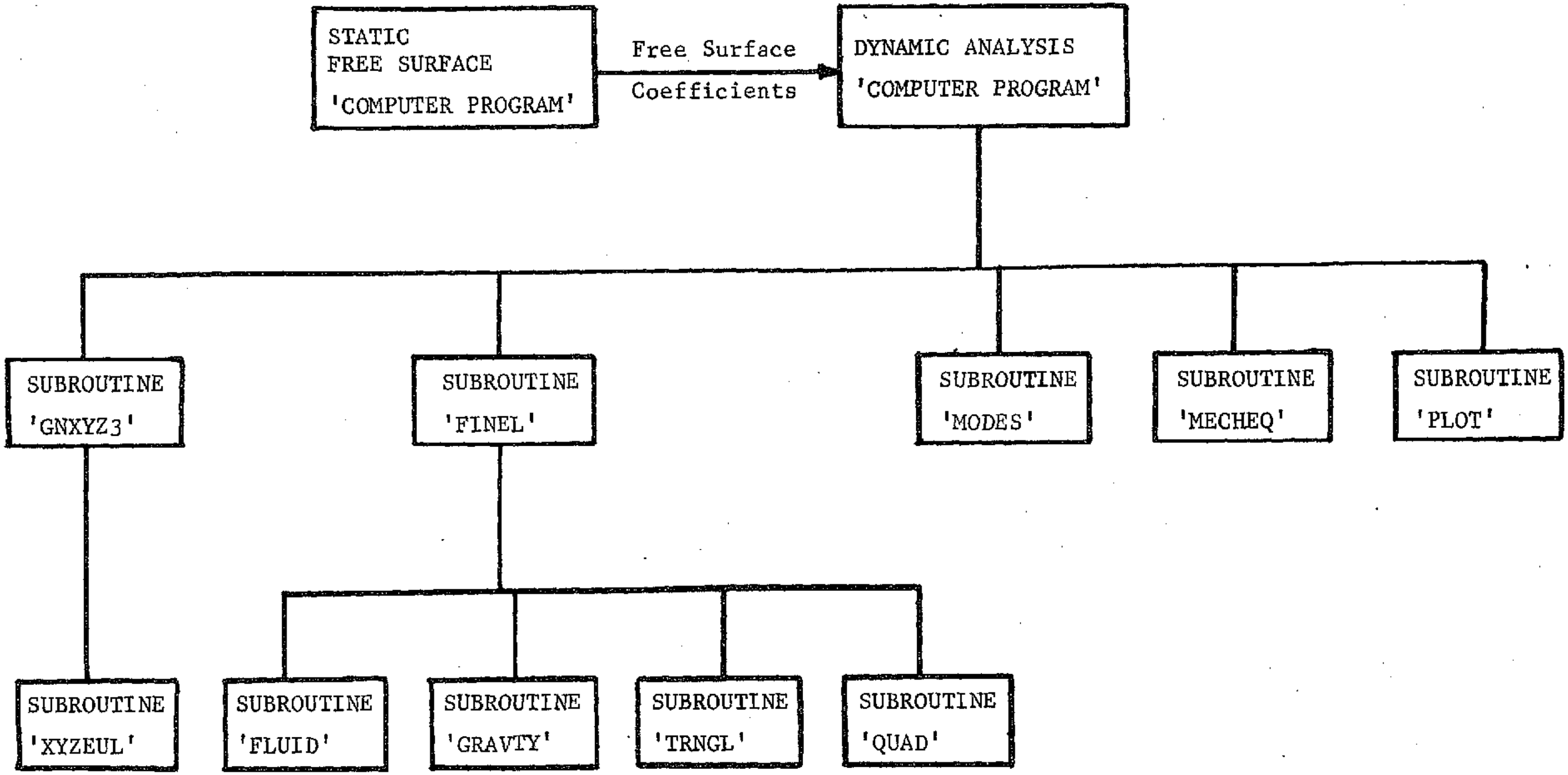
9. "Introduction to the Finite Element Method," by C. S. Desai and John F. Abel, published by Van Nostrand Reinhold Company.

7. APPENDIX

The analysis of the system is accomplished in two main steps. The first is the static free surface definition and the second is the dynamic analysis. The dynamic analysis program includes the automatic generation of joint X, Y, Z values, degree of freedom numbers, Euler angles, element joint numbers, calculation of mode shapes, and frequencies, mechanical equivalent slosh mass and plots. A computer program has been developed for these steps. A schematic flow chart is shown here for the system analysis steps. The listing of the static free surface computer program is given in Appendix A-1 and that of Dynamic analysis computer program in Appendix A-2. The important parameters input to the programs and subroutines are explained in Appendices B-1 and B-2, respectively, using typical input listings.

A brief summary of important subroutine functions are presented in the following pages.

Brief Summary:



Summary of Programs:

STATIC FREE SURFACE	obtains the static free surface
DYNAMIC ANALYSIS	obtains the dynamic characteristics of the system (frequencies of mode shapes): and the mechanical equivalent

Summary of Subroutines (used in the DYNAMIC ANALYSIS Program):

GNXYZ3	input data for 'FINEL'
XYZEUL	automatic generation of input for 'FINEL'
FINEL	generates mass and stiffness matrices
FLUID	generates mass and stiffness for fluid only
GRAVTY	generates gravity contribution to stiffness matrix
TRNGL	generates mass and stiffness for non-fluid triangular elements
QUAD	generates mass and stiffness for non-fluid quadrilateral elements
MODES	obtains frequencies and mode shapes
MECHEQ	obtains mechanical equivalent for the sloshing fluid and bladder
PLOT	plots the mode shapes for the mid-plane

APPENDIX - A1
STATIC FREE SURFACE PROGRAM

```
      PROGRAM SEPS (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      DIMENSION F(15), DFOQ(15,15), C(15,15), RHSV(15), VEC(15),
     1          VEC1(15), DQ(15), C11(15,15), C12(15,15), WORK(15,15),
     2          T(15,15), TT(15,15), C1C2(15,15),PLTIND(83),PLTDEP(83)
C-
      EQUIVALENCE (PLTIND(1),WORK(1)) , (PLTDEP(1),WORK(84))
C-
      E X T E R N A L  FT1,FT2,FT3,FT4,FT5,FT6,FT7
C
      COMMON/BLK1/  E,TH,RHO,G,P,R,NO
      C O M M O N  Q(15)
C
 1001 FORMAT(I5,3D17.8)
 2001 FORMAT(1H1,15(/),25X,*MAXIMUM ITERATION LIMIT REACHED*)
 2002 FORMAT (1X,5D20.10)
C
      DATA KQ/15/
      DATA A/0.0D00/, R/8.0D00/, EPS/1.0D-3/, E/200.0D00/
      DATA TH/0.06D00/, RHO/0.0013D00/, B/8.0D00/
      DATA IN/16/, NC/4/
C-
C     ****************************************************************
C     VARIABLES-
C-    IN = NO. OF INTERVALS FOR NUMERICAL INTEGRATION.
C-    A = LOWER LIMIT
C-    B = UPPER LIMIT
C-    NO = ORDER OF THE POLYNOMIAL
C-    TH = THICKNESS OF THE BLADDER.
C-    E = YOUNGS MODULUS FOR THE BLADDER MATERIAL.
C-    RHO = MASS DENSITY OF THE FLUID.
C-    G = ACCELERATION DUE GRAVITY
C-    P = ULLAGE PRESSURE.
C-    R = RADIUS OF THE BARREL
C-    NC = NO. OF CONSTRANT EQUATIONS
C-    UV = ULLAGE VOLUME
C-    EPS = EPSILON TO COMPARE DQ(I)-S.
C     ****************************************************************
C
C-    ----------------------------------------------------------------
C
C-    INPUTS-
C     NO,G,P,PCT          (I5,3D17.8)
C-    ----------------------------------------------------------------
C
   10 CALL START
      READ(5,1001) NO,G,P,PCT
      CALL ZERO(DFOQ,KQ,KQ,KQ)
      CALL ZERO (C,KQ,KQ,KQ)
      CALL ZERO (VEC,KQ,1,KQ)
      PI = 2.0 * ASIN(1.0)
```

```
      ARCVAL = 1.00 * PI * R / 2.0
      UV=0.5*(1.0-PCT)*PI*R**2
      VEC (NC) = UV
      N1 = NO+1
      N2 = NO + 2
C-    GENERATE THE MATRIX (C)0.
C-    FIRST ROW
      C(1,1)=1.0
      DO 130 J=1,NO
  130 C(1,J+1) = B**J
C-    SECOND ROW
      C(2,1)=0.0
      C(2,2)=1.0
      DO 135 J=2,NO
  135 C(2,J+1) = B**(J-1)*FLOAT(J)
C-    THIRD ROW
      C(3,1)=0.0
      C(3,2)=1.0
      DO 140 J=3,N1
  140 C(3,J)=0.0
C-    FOURTH ROW.
      DO 145 J=1,N1
  145 C(4,J) = (B**J) / FLOAT(J)
C
      CALL WRITE (C,NC,N1,2HC0,KQ)
C-    PARTITION (C) TO MAKE (C11) AND (C12)
C
      DO 150 I=1,NC
      DO 150 J=1,NC
  150 C11(I,J)=C(I,J)
      NC1 = NC + 1
      N1NC=N1-NC
      DO 155 I=1,NC
      DO 155 J=1,N1NC
  155 C12(I,J)=C(I,J+NC)
      CALL WRITE (C11,NC,NC,3HC11,KQ)
      CALL WRITE (C12,NC,N1NC,3HC12,KQ)
C-    GET (C11) INVERSE
      CALL INV1(C11,WORK,NC,KQ)
      CALL WRITE (WORK,NC,NC,6HC11INV,KQ)
C
C-    GET (-C11)INVERSE TIMES (C12)
      CALL MULT (WORK,C12,C1C2,NC,NC,N1NC,KQ,KQ)
      CALL ZERO (T,KQ,KQ,KQ)
      DO 160 I=1,4
      DO 160 J=1,N1NC
  160 T(I,J)=-C1C2(I,J)
      DO 165 I=NC1,N1
      INC1=I-NC
  165 T(I,INC1)=1.0
      CALL WRITE (T,N1,N1NC,4HTRAN,KQ)
C-    FORM MATRIX (C11)INVERSE TIMES THE RIGHT HAND SIDE
```

```
      CALL ZERO (Q,KQ,1,KQ)
      CALL MULT (WORK,VEC,Q,NC,NC,1,KQ,KQ)
      Q(N2) = SIMPS(A,B,IN,FT7,IDUM,IDUM)/ARCVAL - 1.0
      CALL WRITE (Q,N2,1,6HQ(I)-I,KQ)
C
C
C-    LOOP TO DETERMINE Q(I)-S.
      MM = 0
  100 CONTINUE
      CALL MULT(C,Q,WORK,NC,N1,1,KQ,KQ)
      CALL WRITE(WORK,NC,1,6HCONSTR,KQ)
      WRITE(6,2000) MM
2000  FORMAT(* ITERATION NO. *,I3)
      MM = MM +1
      IF (MM.GT.50) GO TO 190
C-    FOR THE FIRST N+1,F-S.
      DO 110 L = 1,N1
  110 F(L) = SIMPS (A,B,IN,FT1,L,IDUM)
C-    FOR THE LAST N+2 ND. F
      CALL WRITE (F,N1,1,2HFN,KQ)
C-    CALCULATE (DFDQ) MATRIX.
      DO 115 I=1,N1
      DO 115 J=1,I
  115 DFDQ(I,J) = SIMPS(A,B,IN,FT3,I,J)
      DO 120 I = 1,N1
      DO 120 J = 1,I
  120 DFDQ(J,I) = DFDQ(I,J)
      CALL WRITE (DFDQ,N1,N1,4HDFDQ,KQ)
C
C-    GET (T) TRANSPOSE AND MULTIPLY
      DO 170 I=1,N1
      DO 170 J=1,N1NC
  170 TT(J,I) = T(I,J)
C
      CALL MULTB (TT,F,N1NC,N1,1,KQ,KQ)
C-    OBTAIN (T) TRANSPOSE X (DFDQ) X (T)
      CALL BTABA (DFDQ,T,N1,N1NC,KQ,KQ)
      CALL WRITE (DFDQ,N1NC,N1NC,6HTTDFQT,KQ)
      CALL INV1(DFDQ,TT,N1NC,KQ)
      CALL MULTB (TT ,F,N1NC,N1NC,1,KQ,KQ)
C-    OBTAIN DELTA-A2
      CALL WRITE (F,N1NC,1,3HDQ2,KQ)
C
C-    GET DELTA-A.
C
      CALL MULT (T,F,DQ,N1,N1NC,1,KQ,KQ)
C
      CALL WRITE (DQ,N1,1,2HDQ,KQ)
C
C-    GET Q(PRESENT) = Q(PREVIOUS)+DQ
C
      DO 175 I=1,N1
```

```
  175 Q(I) = Q(I) - DQ(I)
      Q(N2) = SIMPS(A,B,IN,FT7,IDUM,IDUM)/ARCVAL - 1.0
C
      CALL WRITE (Q,N2,1,5HNQI-S,KQ)
      WRITE (6,2002) (Q(I),I=1,N2)
C
      DO 180 LL=1,N1
      IF (DABS(DQ(LL)).GT.EPS) GO TO 100
  180 CONTINUE
C-    PLOT FREE SURFACE SHAPE
      PLTIND(1) = R
      PLTDEP(1) = 2*R
      PLTIND(83) = 0.0
      PLTDEP(83) = 0.0
      DEL = R / 80.0
      X = A - DEL
      DO 185 I = 2,81
      X = X + DEL
      PLTIND(I) = X
      PLTDEP(I) = Z(X)
  185 CONTINUE
      PLTIND(82) = R
      PLTDEP(82) = Z(R)
      CALL WRITE(PLTIND,83,2,6HSHAPE ,83)
      GO TO 195
  190 WRITE (6,2001)
  195 CONTINUE
      GO TO 10
      END
      DOUBLEPRECISION FUNCTION SIMPS (A,B,N,F,NP,INT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C-    THIS FUNCTION SUPPLIES THE NUMERICALLY INTEGRATED VALUE OF THE
C-    INTEGRAND.
C
      COMMON/BLK1/  E,TH,RHO,G,P,R,NO
      C O M M O N  Q(15)
C
C-    A-LOWER LIMIT
C-    B-UPPER LIMIT
C-    N-NO. OF INTERVALS
C-    F-FUNCTION
C-    NP-GENERALIZED COORDINATE NUMBER (AS TO WHICH ONE IT IS)
C
C-    INITIALIZE PARAMETERS
      TWOH=(B-A)/N
      H = TWOH/2.0
      SUMEND=0.0
      SUMMID = 0.0
C
C-    TWOH-INTERVAL
C-    H-HALF INTERVAL
```

```
C-    SUMMEND-SUM OF F(XI), I BEING EVEN EXCEPTING I=2N
C-    SUMMID-SUM OF F(XI), I BEING ODD.
C
C-    EVALUATE SUMEND AND SUMMID.
      DO 1 K = 1,N
      X=A+FLOAT(K-1)*TWOH
      SUMEND = SUMEND + F(X,NP,INT)
    1 SUMMID = SUMMID + F(X+H,NP,INT)
C
C-    RETURN ESTIMATED VALUE OF THE INTEGRAL.
      SIMPS = (2.0*SUMEND+4.0*SUMMID-F(A,NP,INT)+F(B,NP,INT))*H/3.0
C
      RETURN
      END
      DOUBLEPRECISION FUNCTION FT1(X,NP,IOU)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C-    THIS FUNCTION DEFINES THE INTEGRAND FOR F-S FOR DISCRETE VALUES OF
C     FOR DIFFERENT EQUATION NUMBERS CORRESPONDING TO DIFFERENT Q-S OR A
C     NP = EQUATION NUMBER
C.
      COMMON/BLK1/  E,TH,RHO,G,P,R,NO
      C O M M O N   Q(15)
C
      N2 = NO + 2
C
      IF (X .EQ. 0.0  .OR. NP .LE. 2) GO TO 5
      TFLOT1 = -FLOAT((NP-1)*(NP-2)) * X**(NP-3)
      GO TO 10
    5 TFLOT1 = 0.0
   10 CONTINUE
      IF (X .EQ. 0.0  .OR. NP .LE. 1) GO TO 15
      TFLOT2 = -FLOAT(NP-1) * X**(NP-2)
      GO TO 20
   15 TFLOT2 = 0.0
   20 CONTINUE
C
      TERM1=((E*TH**3)/12.0)*((1.0+ZP(X)**2)**(-3.50))
     1       *(((1.0+ZP(X)**2)*(ZDP(X))* TFLOT1)
     2       -(( 2.50)*(ZDP(X)**2)*(ZP(X))* TFLOT2))
C
      TERM2=(E*TH*Q(N2)**2*0.5)*(((1.0+ZP(X)**2)**(-0.5))
     1       *(ZP(X))* TFLOT2)
C
      IF (X .EQ. 0.0) GO TO 25
      TERM3= (RHO*G)*Z(X)*(-(X**(NP-1)))
C
      TERM4 = P*(X**(NP-1))
C
      GO TO 30
   25 CONTINUE
      TERM3 = 0.0
      TERM4 = 0.0
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
   30 FT1=TERM1+TERM2+TERM3+TERM4
C
      RETURN
      END
      DOUBLEPRECISION FUNCTION FT2(X,NP,IDU)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C-    THIS FUNCTION DEFINES THE INTEGRAND FOR DISCRETE X FOR (NO+2) EQUA
C
      COMMON/BLK1/  E,TH,RHO,G,P,R,NO
      C O M M O N   Q(15)
C
      N2 = NO + 2
C
      TERM=(E*TH*Q(N2))*((1.0+ZP(X)**2)**(0.5))
C
      FT2 = TERM
C
      RETURN
      END
      DOUBLEPRECISION FUNCTION Z(X)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C-    THIS FUNCTION DEFINES THE FUNCTION Z(X) AT DISCRETE X.
C
      COMMON/BLK1/  E,TH,RHO,G,P,R,NO
      C O M M O N   Q(15)
C
      IF (R .EQ. X) GO TO 5
      Z = (R**2-X**2)**0.5+R-W(X)
      GO TO 10
    5 Z = R - W(X)
   10 CONTINUE
C
      RETURN
      END
      DOUBLEPRECISION FUNCTION W(X)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C-    THIS FUNCTION DEFINES W(X) AT DISCRETE VALUE OF X.
C
      COMMON/BLK1/  E,TH,RHO,G,P,R,NO
      C O M M O N   Q(15)
C
      N1 = NO + 1
      W1 = 0.0
      IF (X .EQ. 0.0) GO TO 15
      DO 10 I=1,N1
   10 W1=W1+Q(I)*X**(I-1)
      GO TO 20
   15 W1= W1 + Q(1)
   20 CONTINUE
      W = W1
C
      RETURN
```

```
      END
      DOUBLEPRECISION FUNCTION ZP(X)
      IMPLICIT DOUBLE PRECISION ( A-H,O-Z)
C-    THIS FUNCTION DEFINES (Z) AT DISCRETE X.
C
      COMMON/BLK1/  E,TH,RHO,G,P,R,NO
      C O M M O N   Q(15)
C
      N1 = NO + 1
      TERM1=0.0
      IF (X .EQ. 0.0) GO TO 15
      IF (R .EQ. X) GO TO 20
      DO 10 I=2,N1
   10 TERM1= TERM1+(Q(I)*X**(I-2))*FLOAT(I-1)
      TERM1=TERM1 + X*(R**2-X**2)**(-0.5)
      GO TO 25
   15 TERM1 = Q(2)
      GO TO 25
   20 CONTINUE
      DO 30 I=2,N1
   30 TERM1 = TERM1 + (Q(I) * X**(I-2)) * FLOAT(I-1)
   25 CONTINUE
C
      TERM = -TERM1
C
      ZP=TERM
C
      RETURN
      END
      DOUBLEPRECISION FUNCTION ZOP(X)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C-    THIS FUNCTION DEFINES (Z) FOR DISCRETE X.
C
      COMMON/BLK1/  E,TH,RHO,G,P,R,NO
      C O M M O N   Q(15)
      N1 = NO + 1
C
      TERM1=0.0
C
      IF (X .EQ. 0.0) GO TO 15
      IF (R .EQ. X) GO TO 20
      DO 10 I=3,N1
   10 TERM1=TERM1+(Q(I)*X**(I-3))*FLOAT((I-1)*(I-2))
      TERM1=  TERM1+(R**2-X**2)**(-0.5)
     1        +X**2*(R**2-X**2)**(-1.5)
      GO TO 25
   15 TERM1 = 2.0 * Q(3) + (1.0 / R)
      GO TO 25
   20 CONTINUE
      DO 30 I=3,N1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
   30 TERM1=TERM1+(Q(I)*X**(I-3))*FLOAT((I-1)*(I-2))
   25 CONTINUE
C
      TERM = -TERM1
C
      ZDP=TERM
C
      RETURN
      END
      DOUBLEPRECISION FUNCTION FT3(X,NP,M)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C-    THIS FUNCTION DEFINES THE INTEGRAND FOR DF(K)/DQ(M) FOR DISCRETE X
C-    K=1,2,---(NO+1) AND M=1,2,------,(NO+1). IN THE MATRIX K IS THE ROW
C-    AND M IS THE COLUMN.
C
      COMMON/BLK1/  E,TH,RHO,G,P,R,NO
      C O M M O N  Q(15)
C
      IF (X .EQ. 0.0  .OR. NP .LE. 2) GO TO 5
      TFLOT1 = -FLOAT((NP-1)*(NP-2)) * X**(NP-3)
      GO TO 10
    5 TFLOT1 = 0.0
   10 CONTINUE
      IF (X .EQ. 0.0  .OR. NP .LE. 1) GO TO 15
      TFLOT2 = -FLOAT(NP-1) * X**(NP-2)
      GO TO 20
   15 TFLOT2 = 0.0
   20 CONTINUE
C
      N2 = NO + 2
C
      A1 = 1.0 + ZP(X)**2
      B1 = ZDP(X)
      C1 = TFLOT1
      A2 = ZDP(X)**2
      B2 = ZP(X)
      C2 = 2.50 * TFLOT2
      A3 = ((1.0+ZP(X)**2)**(-0.50))
      B3 = ZP(X)
      C3 = TFLOT2
      D = (1.0 + ZP(X)**2) **(-3.50)
      A1AM = 2.0 * ZP(X) *PZP(X,M)
      B1AM = PZDP(X,M)
      A2AM = 2.0 * ZDP(X) * PZDP(X,M)
      B2AM = PZP(X,M)
      A3AM = -ZP(X) * ((1.0+ZP(X)**2)**(-1.50)) * PZP(X,M)
      B3AM = PZP(X,M)
      DAM = (-7.0) * ((1.0 + ZP(X)**2) ** (-4.50)) * ZP(X) * PZP(X,M)
C
      TERM1 = ((E*TH**3)/12.0) *
     1         (D * ((A1*C1*B1AM) + (B1*C1*A1AM) - (A2*C2*B2AM) -
     2         (B2*C2*A2AM)) + (A1*B1*C1 - A2*B2*C2) * DAM)
C
```

```
      TERM2 = (0.50*E*TH*Q(N2)**2) *
     1         (A3*C3*B3AM + B3*C3*A3AM)
C
      IF (X .EQ. 0.0) GO TO 25
      TERM3=(RHO*G)*(PZ(X,M)*(-(X**(NP-1))))
      GO TO 30
   25 TERM3 = 0.0
C
   30 TERM= TERM1+TERM2+TERM3
C
      FT3=TERM
C
      RETURN
      END
      DOUBLEPRECISION FUNCTION FT4(X,NP,M)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C-    THIS FUNCTION DEFINESINTEGRAND FOR DF(K)/DQ(M) FOR DISCRETE X AND
C     K=1,2,---(NO+1) AND M=NO+2
C
      COMMON/BLK1/  E,TH,RHO,G,P,R,NO
      C O M M O N   Q(15)
C
      N2 = NO + 2
      IF (X .EQ. 0.0  .OR. NP .LE. 1) GO TO 15
      TFLOT2 = -FLOAT(NP-1) * X**(NP-2)
      GO TO 20
   15 TFLOT2 = 0.0
   20 CONTINUE
C
C
      TERM=(E*TH*Q(N2))*((1.0+ZP(X)**2)**(-0.5)*ZP(X) * TFLOT2)
C
      FT4 = TERM
C
      RETURN
      END
      DOUBLEPRECISION FUNCTION FT5(X,NP,M)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C-    THIS FUNCTION DEFINES INTEGRAND FOR DF(K)/DQ(M) FOR DISCRETE X AND
C-    K=NO+2 AND M=1,2,----NO+1.
C
      COMMON/BLK1/  E,TH,RHO,G,P,R,NO
      C O M M O N   Q(15)
C
      N2 = NO + 2
      TERM=(E*TH*Q(N2))*((0.5)*(1.0+ZP(X)**2)**(-0.5)
     1      *(2.0*ZP(X)*PZP(X,M)))
C
      FT5 = TERM
C
      RETURN
      END
```

```
      DOUBLEPRECISION FUNCTION FI6(X,NP,M)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C-    THIS FUNCTION DEFINES INTEGRAND FOR DF(K)/DQ(M) FOR DISCRETE X AND
C-    K=NO+2 AND M=NO+2
C
      COMMON/BLK1/  E,TH,RHO,G,P,R,NO
      C O M M O N  Q(15)
C
      TERM=(E*TH)*(1.0+ZP(X)**2)**(0.5)
C
      FI6=TERM
C
      RETURN
      END
      DOUBLEPRECISION FUNCTION PZDP(X,M)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C-    THIS FUNCTION CALCULATES THE VALUE OF Q(Z) OR PZDP FOR DISCRETE X.
      IF (X .EQ. 0.0) GO TO 10
      IF (M .GT. 2) GO TO 5
      TERM = 0.0
      GO TO 15
    5 CONTINUE
      TERM = -FLOAT((M-1)*(M-2)) * X**(M-3)
      GO TO 15
   10 CONTINUE
      TERM = 0.0
      IF (M .EQ. 3) TERM = -2.0
   15 CONTINUE
      PZDP = TERM
C
      RETURN
      END
      DOUBLEPRECISION FUNCTION PZP(X,M)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C-    THIS FUNCTION CALCULATES THE VALUE OF Q (Z) OR PZP(X) FOR DISCRETE
      IF ( X .EQ. 0.0) GO TO 10
      IF (M .GT. 1) GO TO 5
      TERM = 0.0
      GO TO 15
    5 CONTINUE
      TERM = -FLOAT(M-1) * X**(M-2)
      GO TO 15
   10 CONTINUE
      TERM = 0.0
      IF (M .EQ. 2) TERM = -1.0
   15 CONTINUE
      PZP = TERM
C
      RETURN
      END
      DOUBLEPRECISION FUNCTION PZ(X,M)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

```
C-     THIS FUNCTION CALCULATES THE VALUE OF 0(Z)ORPZ(X) FOR DISCRETE X.
       IF (X .EQ. 0.0) GO TO 5
       TERM = -(X**(M-1))
       GO TO 10
     5 CONTINUE
       TERM = 0.0
       IF (M .EQ. 1) TERM = -1.0
    10 CONTINUE
       PZ = TERM
C
       RETURN
       END
       DOUBLEPRECISION FUNCTION FT7(X,IDUM,JDUM)
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C-      FUNCTION TO EVALUATE ARC LENGTH OF DEFORMED BLADDER
C
       FT7 = DSQRT(1.0 + ZP(X)*ZP(X))
       RETURN
       END
       SUBROUTINE PAGEHD
       COMMON/LSTART/IRUNNO,IDATE,NPAGE,UNAME(3),TITLE1(12),TITLE2(12)
       DATA  NIT,NOT/5,6/
C
  2001 FORMAT (9H1RUN NO. ,A6,42X,5HDATE ,A6/
      *55X,7HRUN BY ,3A6/102X,A9,12H CLOCK  TIME/10X,12A6,19X,F10.3,
      *12H SEC. CPTIME/10X,12A6)
C
       CALL TIME   (DTIME)
       CALL SECOND (CP)
       WRITE (NOT,2001) IRUNNO,IDATE,UNAME,DTIME,
      *                 TITLE1,CP,TITLE2
C
       RETURN
       END
       SUBROUTINE MULTB (A,BZ,NRA,NRB,NCB,KA,KBZ)
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       DIMENSION A(KA,1),BZ(KBZ,1)
       COMMON /LWRKV1/ W(40)
C
C  MATRIX MULTIPLICATION.  A * B = Z.
C  USES TWO WORK SPACES. RESULT (Z) IS PLACED IN B.
C  BZ MUST BE DIMENSIONED LARGE ENOUGH IN MAIN PROGRAM TO CONTAIN THE
C  LARGER OF B OR Z.
C  CALLS FORMA SUBROUTINE ZZBOMB.
C  THE MAXIMUM SIZE IS
C     NRB = 40
C  DEVELOPED BY CARL BODLEY.  JANUARY 1965.
C  LAST REVISION BY R L WOHLEN.  JULY 1972.
C
C     SUBROUTINE ARGUMENTS
C  A   = INPUT  MATRIX. SIZE(NRA,NRB).
```

```
C  BZ   = INPUT  MATRIX. SIZE(NRB,NCB).
C       = OUTPUT RESULT MATRIX. SIZE(NRA,NCB).
C  NRA = INPUT  NUMBER OF ROWS OF MATRICES A,Z.
C  NRB = INPUT  NUMBER OF ROWS OF MATRIX B, COLS OF MATRIX A. MAX=40.
C  NCB = INPUT  NUMBER OF COLS OF MATRICES B,Z.
C  KA  = INPUT  ROW DIMENSION OF A IN CALLING PROGRAM.
C  KBZ = INPUT  ROW DIMENSION OF BZ IN CALLING PROGRAM.
C
                                                           NERROR=1
      IF (NRB.GT.40 .OR. NRA.GT.KBZ .OR. NRB.GT.KBZ) GO TO 999
C
      DO 40 J=1,NCB
      DO 20 K=1,NRB
   20 W(K) = BZ(K,J)
      DO 40 I=1,NRA
      S = 0.
      DO 30 K=1,NRB
   30 S = S + A(I,K)*W(K)
   40 BZ(I,J) = S
      RETURN
C
  999 CALL ZZBOMB (6HMULTB ,NERROR)
      END
      SUBROUTINE ZZBOMB (SUBNAM,NERROR)
      DIMENSION DFMSSG(8)
      DATA NIT,NOT/5,6/
C
C  CONTROL A COMPUTER RUN AFTER AN ERROR MESSAGE HAS BEEN ENCOUNTERED
C  IN ANY OF THE FORMA SUBROUTINES.
C  ON THE CDC 6000 SERIES COMPUTER THIS INVOLVES ...
C     (1) PRINT ERROR MESSAGE, INCLUDING SUBNAM AND NERROR, IN PRINTOUT.
C     (2) PRINT ERROR MESSAGE, INCLUDING SUBNAM AND NERROR, IN DAYFILE.
C     (3) CALL TO NON-EXISTANT ROUTINE TO CAUSE ABNORMAL
C         STOP AND TRANSFER TO THE EXIT CARD.
C  CODED BY  RL WOHLEN.  SEPTEMBER 1970.
C  LAST REVISION BY R HRUDA,  JAN 1974.
C
C     SUBROUTINE ARGUMENTS
C  SUBNAM = INPUT SUBROUTINE NAME IN WHICH ERROR OCCURRED.
C  NERROR = INPUT ERROR NUMBER FROM SUBROUTINE WHERE ERROR OCCURRED.
C
 3001 FORMAT (1H1)
 3002 FORMAT (19H ZZBOMB - ROUTINE (,A6,11H), NERROR (,I3,1H))
C
      WRITE (NOT,3001)
      WRITE (NOT,3002) SUBNAM,NERROR
      ENCODE (40,3002,DFMSSG) SUBNAM,NERROR
      CALL REMARK (DFMSSG)
      CALL ABNORML
C
      END
      SUBROUTINE BTABA (AZ,B,NRB,NCB,KAZ,KB)
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION AZ(KAZ,1), B(KB,1)
      COMMON / LWRKV1 / W(40)
C
C   TRIPLE MATRIX PRODUCT.  B(TRANSPOSE) * A * B = Z.
C   A MUST BE SYMMETRIC TO GET CORRECT ANSWER.
C   Z WILL BE SYMMETRIC. UPPER HALF CALCULATED, REFLECTED TO LOWER HALF.
C   USES TWO WORK SPACES. RESULT (Z) IS PLACED IN A.
C   AZ MUST BE DIMENSIONED LARGE ENOUGH IN MAIN PROGRAM TO CONTAIN THE
C   LARGER OF A OR Z.
C   CALLS FORMA SUBROUTINE ZZBOMB.
C   THE MAXIMUM SIZES ARE
C      NRB = 40
C      NCB = 40
C   DEVELOPED BY W A BENFIELD.  MAY 1972.
C   LAST REVISION BY R A PHILIPPUS.  JUNE 1972.
C
C      SUBROUTINE ARGUMENTS
C   AZ  = INPUT  INNER MATRIX. SIZE(NRB,NRB).
C       = OUTPUT RESULT MATRIX. SIZE(NCB,NCB).
C   B   = INPUT  OUTER MATRIX. SIZE(NRB,NCB).
C   NRB = INPUT  NUMBER OF ROWS OF MATRIX B, SIZE OF MATRIX A. MAX=40.
C   NCB = INPUT  NUMBER OF COLS OF MATRIX B, SIZE OF MATRIX Z. MAX=40.
C   KAZ = INPUT  ROW DIMENSION OF AZ IN CALLING PROGRAM.
C   KB  = INPUT  ROW DIMENSION OF B IN CALLING PROGRAM.
C
                                                              NERROR=1
      IF (NRB.GT.40 .OR. NCB.GT.40 .OR. NRB.GT.KAZ .OR. NCB.GT.KAZ)
     *   GO TO 999
C
      DO 20 I=1,NRB
      DO 5 K=1,NRB
    5 W(K) = AZ(I,K)
      DO 20 J=1,NCB
      S = 0.0
      DO 10 K=1,NRB
   10 S = S + W(K)*B(K,J)
   20 AZ(I,J) = S
C
      DO 30 J=1,NCB
      DO 25 I=1,J
      W(I) = 0.0
      DO 25 K=1,NRB
   25 W(I) = W(I)+B(K,I)*AZ(K,J)
      DO 30 I=1,J
      AZ(I,J) = W(I)
   30 AZ(J,I) = W(I)
      RETURN
C
  999 CALL ZZBOMB (6HBTABA ,NERROR)
      END
      SUBROUTINE INV1  (A,Z,N,KR)
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION  A(1), Z(1)
      COMMON /LWRKV1/ G(20), DETR(20)
      COMMON /LWRKV2/ IX(20), B(20)
      DATA NIT,NOT/5,6/
C
C  MATRIX INVERSION (A**-1 = Z).   BORDERING METHOD.
C  THE DETERMINANT RATIO DET(I+1) / DET(I) IS PRINTED. DET(I) IS THE
C  DETERMINANT OF THE FIRST I BY I SUB-MATRIX OF A.
C  THE INVERSION CHECK Z*A IS CALCULATED AND PRINTED.
C  MATRICES A,Z MAY SHARE SAME CORE LOCATIONS. (Z*A CHECK IS INVALID).
C  CALLS FORMA SUBROUTINES PAGEHD,ZZBOMB.
C  THE MAXIMUM SIZE IS
C     N = 20
C  DEVELOPED BY   BOB DILLON.  FEBRUARY 1965.
C  LAST REVISION BY J ERNST,  OCT 1973.
C
C     SUBROUTINE ARGUMENTS
C  A  = INPUT  MATRIX TO BE INVERTED. SIZE(N,N).
C  Z  = OUTPUT RESULT MATRIX. SIZE(N,N).
C  N  = INPUT  SIZE OF MATRICES A,Z. MAX=20.
C  KP = INPUT  ROW DIMENSION OF A,Z IN CALLING PROGRAM.
C
 2000 FORMAT (// 10X,10(7X,1H(,I2,1H)))
 2001 FORMAT (// 10X,45HSUBROUTINE INV1 HAS CALCULATED THE DATA BELOW
     *          ///10X,44HTHE DETERMINANT RATIOS DET(I+1) / DET(I) ARE
     *          // (13X,10D11.3))
 2002 FORMAT (///10X,37HTHE (A**-1)*(A) INVERSION CHECK GIVES
     *          ///10X,25HTHE DIAGONAL ELEMENTS ARE // (13X,8D14.6))
 2003 FORMAT (// 10X,35HTHE MAXIMUM OFF-DIAGONAL ELEMENT IS
     *                         D11.3, 2X, 4HAT ( I3, 1H, I3, 1H) )
C
                                                        NERROR=1
      IF (N .GT. 20) GO TO 999
C
      DO 160 I=2,N
  160 IX(I) = I
C  INVERT FIRST NON-ZERO ELEMENT IN FIRST COLUMN.
      DO 190 I=1,N
      IF (A(I) .NE. 0.) GO TO 220
  190 CONTINUE
                                                        NERROR=2
      GO TO 999
C
C  START INVERSION WITH ROW I.
  220 DETR(1) = A(I)
      Z(I) = 1. / A(I)
      IF (N .EQ. 1) RETURN
C
      IX(I) = 1
      IX(1) = I
C  BORDERING LOOP.
```

```
      DO 630 L=2,N
           K = L
           L1 = L - 1
  250 S = 0.
                        MIXL = KR * (IX(L) - 1)
                        LL = IX(L) + MIXL
      DO 450 I=1,L1
                        MIXI = KR * (IX(I) - 1)
                        LI = IX(L) + MIXI
      B(I) = 0.
      G(I) = 0.
      DO 440 J=1,L1
                        MIXJ = KR * (IX(J) - 1)
                        IJ = IX(I) + MIXJ
                        JL = IX(J) + MIXL
      B(I) = B(I) - Z(IJ)* A(JL)
                        JI = IX(J) + MIXI
                        LJ = IX(L) + MIXJ
  440 G(I) = G(I) - A(LJ)* Z(JI)
  450 S = S + A(LI)* B(I)
      AL = A(LL)+ S
      IF (A(LL) .EQ. 0.) GO TO 480
      ALBAR = DABS (AL / A(LL))
      GO TO 490
  480 ALBAR = DABS (AL)
  490 IF (ALBAR .GE. .1D-6) GO TO 550
C
C  INTERCHANGE ROWS AND COLUMNS.
      K = K + 1
      IF (K .GT. N) GO TO 540
      IX L = IX(L)
      IX(L) = IX(K)
      IX(K) = IX L
      GO TO 250
  540 IF (ALBAR .GE. .1D-8) GO TO 550
                                                             NERROR=3
      GO TO 999
C
  550 Z(LL)= 1. / AL
      DETR(L) = AL
      DO 570 I=1,L1
                        IL = IX(I) + MIXL
      LI = IX(L) + KR * (IX(I) - 1)
      Z(IL)= B(I) * Z(LL)
      Z(LI)= G(I) * Z(LL)
      DO 570 J=1,L1
      IJ = IX(I) + KR * (IX(J) - 1)
  570 Z(IJ)= Z(IJ)+ G(J) * Z(IL)
  630 CONTINUE
C
C  COMPUTE INVERSION CHECK Z*A.
      XOFF = 0.0
```

```
      DO 720 I=1,N
      DO 710 J=1,N
      X = 0.0
      KJA = KR * (J-1)
      DO 703 K=1,N
      IK = I + KR*(K-1)
      KJ = K + KJA
  703 X = X + Z(IK) * A(KJ)
      IF (I .NE. J) GO TO 705
      G(I) = X
      GO TO 710
  705 IF (DABS(X) .LT. DABS(XOFF)) GO TO 710
      XOFF = X
      IOFF = I
      JOFF = J
  710 CONTINUE
  720 CONTINUE
C
C  PRINT THE DETERMINANT RATIO AND INVERSION CHECK.
      CALL PAGEHD
      WRITE (NOT,2000) (JC, JC=1,10)
      WRITE (NOT,2001) (DETR(I), I=1,N)
      WRITE (NOT,2002) ( G  (I), I=1,N)
      WRITE (NOT,2003) XOFF,IOFF,JOFF
      RETURN
C
  999 CALL ZZBOMB (6HINV1   ,NERROR)
      END
      SUBROUTINE MULT   (A,B,Z,NRA,NRB,NCB,KRA,KRB)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION  A(KRA,1), B(KRB,1), Z(KRA,1)
C
C  MATRIX MULTIPLICATION.  A * B = Z.
C  DEVELOPED BY R L WOHLEN.  FEBRUARY 1965.
C  LAST REVISION BY R L WOHLEN.  JULY 1972.
C
C      SUBROUTINE ARGUMENTS
C  A   = INPUT  MATRIX. SIZE(NRA,NRB).
C  B   = INPUT  MATRIX. SIZE(NRB,NCB).
C  Z   = OUTPUT RESULT MATRIX. SIZE(NRA,NCB).
C  NRA = INPUT  NUMBER OF ROWS OF MATRICES A,Z.
C  Z   = OUTPUT RESULT MATRIX. SIZE(NRA,NCB).
C  NRB = INPUT  NUMBER OF ROWS OF MATRIX B, COLS OF MATRIX A.
C  NCB = INPUT  NUMBER OF COLS OF MATRICES B,Z.
C  KRA = INPUT  ROW DIMENSION OF A,Z IN CALLING PROGRAM.
C  KRB = INPUT  ROW DIMENSION OF B IN CALLING PROGRAM.
C
      DO 20 I=1,NRA
      DO 20 J=1,NCB
      S = 0.
      DO 10 K=1,NRB
   10 S = S + A(I,K)*B(K,J)
   20 Z(I,J) = S
```

```
      RETURN
      END
      SUBROUTINE MULTA (AZ,B,NRA,NRB,NCB,KAZ,KB)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION AZ(KAZ,1), B(KB,1)
      COMMON / LWRKV1 / W(40)
C
C  MATRIX MULTIPLICATION. A * B = Z.
C  USES TWO WORK SPACES. RESULT (Z) IS PLACED IN A.
C  AZ MUST BE DIMENSIONED LARGE ENOUGH IN MAIN PROGRAM TO CONTAIN THE
C  LARGER OF A OR Z.
C  CALLS FORMA SUBROUTINE ZZBOMB.
C  THE MAXIMUM SIZE IS
C     NRB = 500
C  DEVELOPED BY C S BODLEY.  JANUARY 1965.
C  LAST REVISION BY R F HRUDA.  JUNE 1972.
C
C     SUBROUTINE ARGUMENTS
C  AZ  = INPUT  MATRIX. SIZE(NRA,NRB).
C      = OUTPUT RESULT MATRIX. SIZE(NRA,NCB).
C  B   = INPUT  MATRIX. SIZE(NRB,NCB)
C  NRA = INPUT  NUMBER OF ROWS OF MATRICES A,Z.
C  NRB = INPUT  NUMBER OF ROWS OF MATRIX B, COLS OF MATRIX A. MAX=500.
C  NCB = INPUT  NUMBER OF COLS OF MATRICES B,Z.
C  KAZ = INPUT  ROW DIMENSION OF AZ IN CALLING PROGRAM.
C  KB  = INPUT  ROW DIMENSION OF B IN CALLING PROGRAM.
C
                                                           NERROR=1
      IF (NRB .GT. 500) GO TO 999
C
      DO 40 I=1,NRA
      DO 20 K=1,NRB
   20 W(K) = AZ(I,K)
      DO 40 J=1,NCB
      S = 0.0
      DO 30 K=1,NRB
   30 S = S + W(K)*B(K,J)
   40 AZ(I,J) = S
      RETURN
C
  999 CALL ZZBOMB (6HMULTA ,NERROR)
      END
      SUBROUTINE WRITE  (A,NR,NC,INAME,KR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(KR,1)
      DATA NIT,NOT/5,6/
C
C  WRITE MATRIX OF REAL NUMBERS ON PAPER.
C  REQUIRES 123 COLUMN (MINIMUM) PRINTER.
C  UP TO 10 DATA FIELDS PER LINE. PRINTS ONLY NON-ZERO FIELD ROWS.
C  CALLS FORMA SUBROUTINE PAGEHD.
C  CODED BY  RL WOHLEN.  DECEMBER 1968.
```

```
C  LAST REVISION BY R HRUDA,  NOV 1973.
C
C      SUBROUTINE ARGUMENTS (ALL INPUT)
C  A      = MATRIX TO BE PRINTED. SIZE(NR,NC).
C  NR     = NUMBER OF ROWS IN MATRIX A.
C  NC     = NUMBER OF COLS IN MATRIX A.
C  INAME = MATRIX IDENTIFICATION. (A6 FORMAT).
C  KR     = ROW DIMENSION OF A IN CALLING PROGRAM.
C
 2010 FORMAT (//15H OUTPUT MATRIX A6,2X 1H(I4,2H X I4,2H ) //
     *          10X,10(7X,1H( I2,1H))/)
 2020 FORMAT (//15H OUTPUT MATRIX A6,2X 1H(I4,2H X I4,2H )
     *          3X, 9HCONTINUED //10X,10(7X,1H( I2,1H))/)
 2030 FORMAT (1X,2I5,2X,  10D11.3)
 2040 FORMAT (14H0END OF WRITE.)
C
C  PULL UP A NEW PAGE FOR MATRIX AND PRINT MATRIX NAME.
      CALL PAGEHD
      WRITE (NOT,2010) INAME,NR,NC,(L,L=1,10)
      NLINE = 0
C
      DO 60 I=1,NR
      NZERO = 0
      JS = 1
   10 JE = JS+9
      IF (JE .GT. NC) JE=NC
C  SEE IF ELEMENTS ARE ZERO.
      DO 20 J=JS,JE
      IF (A(I,J) .NE. 0.) GO TO 30
   20 CONTINUE
      GO TO 40
   30 NLINE = NLINE+1
      IF (NLINE .LE. 44) GO TO 35
      CALL PAGEHD
      WRITE (NOT,2020) INAME,NR,NC,(L,L=1,10)
      NLINE = 1
   35 WRITE (NOT,2030) I,JS,(A(I,J), J=JS,JE)
      NZERO = 1
   40 IF (JE .EQ. NC) GO TO 50
      JS = JS+10
      GO TO 10
C  SKIP A SPACE BETWEEN EACH ROW IF THERE ARE MORE THAN 10 COLUMNS
C  AND SOMETHING HAS BEEN WRITTEN.
   50 IF (NC.LE.10 .OR. NZERO.EQ.0 .OR. I.EQ.NR)  GO TO 60
      NLINE = NLINE+1
      WRITE (NOT,2030)
   60 CONTINUE
C
      WRITE (NOT,2040)
      RETURN
      END
```

```
      SUBROUTINE ZERO(Z,NR,NC,KR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION  Z(KR,1)
C
C  GENERATE A MATRIX OF ZEROES.
C  CODED BY  RL WOHLEN.  FEB 1965.
C
C     SUBROUTINE ARGUMENTS
C  Z  = OUTPUT MATRIX GENERATED. SIZE(NR,NC).
C  NR = INPUT  NUMBER OF ROWS IN MATRIX Z.
C  NC = INPUT  NUMBER OF COLS IN MATRIX Z.
C  KR = INPUT  ROW DIMENSION OF MATRIX Z IN CALLING PROGRAM.
C
      DO 10 I=1,NR
      DO 10 J=1,NC
   10 Z(I,J) = 0.0
      RETURN
      END
      SUBROUTINE START
      DIMENSION MONTHN(12),MONTHL(12)
      COMMON /LSTART/IRUNNO,IDATE,NPAGE,UNAME(3),TITLE1(12),TITLE2(12)
      DATA NIT,NOT/5,6/
      DATA MONTHN/2H01,2H02,2H03,2H04,2H05,2H06,
     *            2H07,2H08,2H09,2H10,2H11,2H12/,
     *     MONTHL/2HJA,2HFE,2HMR,2HAP,2HMY,2HJN,
     *            2HJL,2HAU,2HSE,2HOC,2HNO,2HDE/
C
C  READS INPUT CARD 1 FOR IRUNNO, UNAME.
C  CHECKS IRUNNO FOR STOP (I.E. IF IRUNNO = STOP, PROGRAM WILL
C  BE STOPPED). YOU SHOULD HAVE A STOP CARD (THE WORD STOP PUNCHED
C  STARTING IN COLUMN 1) AFTER YOUR REGULAR DATA DECK.
C  IF IRUNNO IS NOT EQUAL TO STOP, THE SUBROUTINE CONTINUES AS FOLLOWS.
C  READS INPUT CARD 2 FOR TITLE1.
C  READS INPUT CARD 3 FOR TITLE2.
C  SETS NPAGE = 0.
C  INTEROGATES COMPUTER TO DEFINE DATE AS AN A6.
C  INTEROGATES MACHINE FOR THE TIME OF DAY AND THE CPTIME
C  AND PRINTS THESE ITEMS ON A SHEET OF THE OUTPUT EVERY
C  TIME THIS ROUTINE IS CALLED.
C
C     INPUT ORDER
C  IRUNNO,UNAME       FORMAT (A6, 4X 3A6)
C  TITLE1             FORMAT (12A6)
C  TITLE2             FORMAT (12A6)
C
C     DEFINITIONS
C  IRUNNO = RUN NUMBER. (A6 FORMAT)
C  IDATE  = DATE. (A6 FORMAT).
C  NPAGE  = PAGE NUMBER.
C  UNAME  = USERS NAME. (3A6 FORMAT)
C  TITLE1 = FIRST  TITLE. (12A6 FORMAT)
C  TITLE2 = SECOND TITLE. (12A6 FORMAT)
C
C  DEDICATED TO G. MOROSOW.
```

```
C  CALLS CDC-6000-SERIES COMPUTER STANDARD ROUTINES  DATE, SECOND, TIME.
C  CALLS MARTIN MARIETTA SPECIAL ROUTINE  PPTIM.
C  CODED FOR YOUR CONVENIENCE BY YOUR FRIENDLY METHODS GROUP.  DEC 1968.
C  LAST REVISION BY R HRUDA,  MAY 1974.
C
 1001 FORMAT (A6, 4X 3A6)
 1002 FORMAT (12A6)
 2002 FORMAT (1H1 6(/) 55X 10HTIME SHEET / 38X 45(1H-) //
     *          38X 30HCURRENT TIME OF DAY IN H,M,S = A10 //
     *          38X 26HTOTAL CPTIME USED TO NOW = F10.3, 9H SECONDS. /
     *          38X 26HTOTAL PPTIME USED TO NOW = I6,4X, 9H SECONDS. )
 2003 FORMAT (36H1END OF INPUT DATA HAS BEEN REACHED.)
 5001 FORMAT (3(1X,A2))
 5002 FORMAT (3A2)
C
      CALL TIME (DTIME)
      CALL SECOND (CTIME)
      WRITE (NOT,2002) DTIME,CTIME,IPTIME
      CALL DATE (IDATE)
      DECODE (9,5001,IDATE)  IM,ID,IY
      DO 20 I=1,12
      IF (IM.EQ.MONTHN(I)) GO TO 30
   20 CONTINUE
   30 IM = MONTHL(I)
      ENCODE (6,5002,IDATE)  ID,IM,IY
C
      READ (NIT,1001) IRUNNO,UNAME
      IF (IRUNNO .NE. 4HSTOP) GO TO 10
      WRITE (NOT,2003)
      STOP
C
   10 READ (NIT,1002) TITLE1
      READ (NIT,1002) TITLE2
      NPAGE = 0
      RETURN
      END
```

APPENDIX - A2
DYNAMIC ANALYSIS PROGRAM

```
SIRISHP00000*F2.MAIN
C
C    OVERLAY MAIN PROGRAM TO CALCULATE SHUTTLE EXTERNAL TANK SLOSH MODES.
C    DEVELOPED BY W BENFIELD, C BODLEY, R PHILIPPUS, R WOHLEN.  JULY 1973.
C    LAST REVISION BY R A PHILIPPUS.  JULY 1974.
C
C    ******************************************************************
C    INPUT DATA READ IN THIS PROGRAM.
C    10 CALL START
C       CALL COMENT
C       IFINIT,TAPEID                                       FORMAT (2A6)
C    15 IOPT                                                FORMAT (A6)
C       IF (IOPT .EQ. 6HSTART ) GO TO 10
C       IF (IOPT .EQ. 6HGNXYZ ) CALL GNXYZ2 (SEE SUBRT FOR INPUT)
C       IF (IOPT .EQ. 6HFINEL ) CALL FINEL  (INPUT DATA FROM GNXYZ2)
C       IF (IOPT .EQ. 6HXTRAMK) CALL PXTRA  (SEE SUBRT FOR INPUT)
C       IF (IOPT .EQ. 6HMODES ) CALL GNIMD  (SEE SUBRT FOR INPUT)
C      *IF (IOPT .EQ. 6HMODES ) CALL OYMODE (SEE SUBRT FOR INPUT)
C       IF (IOPT .EQ. 6HREDUCE) CALL REDUCE (SEE SUBRT FOR INPUT)
C       IF (IOPT .EQ. 6HSAVEMK) CALL SAVE   (SEE SUBRT FOR INPUT)
C       IF (IOPT .EQ. 6HSUBSTR) CALL SUBSTR (SEE SUBRT FOR INPUT)
C       IF (IOPT .EQ. 6HPLOT  ) CALL SPLT1  (SEE SUBRT FOR INPUT)
C       GO TO 15
C
      COMMON / RWTAPS / NUTEL,NUTXYZ,NUTLT,NUTST,NUTMX,NUTKX,NUTBX
      COMMON / BTAPEF / NUTMF,NUTKF,NUT1F,NUT2F,NUT3F
      COMMON / BTAPET / NUTMT,NUTKT,NUT1T,NUT2T,NUT3T,NUT4T,NUT5T
      COMMON / BTAPEM / NUTMM,NUTKM,NUTTM,NUTPM,NUTFM,NUT1M,NUT2M,NUT3M,
     *                  NUT4M,NUT5M,NUT6M,NUT7M
      COMMON / BTAPED / NUTKD,NUTLD,NUTDD,NUT1D,NUT2D,NUT3D,NUT4D
      COMMON / BTAPEB / NUTKB,NUTBB,NUTPB,NUT1B,NUT2B,NUT3B,NUT4B,
     *                  NUT5B,NUT6B,NUT7B
      COMMON / BTAPEC / NUTMC,NUTKC,NUTTC,NUT1C,NUT2C,NUT3C,NUT4C,NUT5C,
     *                  NUT6C
      COMMON / BTAPER / NUTMR,NUTKR,NUTTR,NUT1R,NUT2R,NUT3R,NUT4R,NUT5R,
     *                  NUT6R
      COMMON / BTAPES / NUTMS,NUTKS,NUTTS
      COMMON / BTAPEP / NUTMP,NUTKP,NUTTP,NUT1P,NUT2P,NUT3P,NUT4P,NUT5P
      COMMON / BTAPEA / NUTPA,NUTFA,NUT1A,NUT2A,NUT3A,NUT4A,NUT5A,
     *                  NUT6A,NUT7A
      COMMON / RESTAP / NRSVT1
      COMMON / RTRANS / IFTRAN
C
      DATA NIT,NOT/5,6/
C    DEFINE READ,WRITE TAPE UNITS FOR ALL OVERLAYS.
      DATA NUTEL,NUTXYZ /
     *        29,    30 /
      DATA NUTLT,NUTST,NUTMX,NUTKX,NUTBX /
     *        31,    1,    2,   26,   27 /
C    DEFINE BUFFER IN,OUT TAPE UNITS FOR FINELO.
      DATA NUTMF,NUTKF,NUT1F,NUT2F,NUT3F/
     *        21,   22,   11,   12,   13/
C    DEFINE BUFFER IN,OUT TAPE UNITS FOR EXTRA M,K OVERLAY.
      DATA NUTMT,NUTKT,NUT1T,NUT2T,NUT3T,NUT4T,NUT5T /
     *        21,   22,   11,   12,   13,   14,   15 /
C    DEFINE BUFFER IN,OUT TAPE UNITS FOR YMODE2.
      DATA NUTMM,NUTKM,NUTPM,NUTTM,NUTFM,NUT1M,NUT2M,NUT3M,NUT4M,NUT5M/
```

```
     *              21,    22,    23,    24,    25,    11,    12,    13,    14,    15/
      DATA NUT6M/
     *              16/
      DATA NUT7M/
     *              17/
C  DEFINE BUFFER IN,OUT TAPE UNITS FOR STATIC DEFLECTION CALCULATIONS.
      DATA NUTKD,NUTLD,NUTDD,NUT1D,NUT2D,NUT3D,NUT4D /
     *              22,    21,    23,    11,    12,    13,    14 /
C  DEFINE BUFFER IN,OUT TAPE UNITS FOR BUCKLING LOAD CALCULATIONS.
      DATA NUTKB,NUTBB,NUTPB,NUT1B,NUT2B,NUT3B,NUT4B,NUT5B,NUT6B,NUT7B/
     *              22,    21,    23,    11,    12,    13,    14,    15,    16,    17/
C  DEFINE BUFFER IN,OUT TAPE UNITS FOR CONSTANT VOLUME FLUID ELEMENT
C  CALCULATIONS.
      DATA NUTMC,NUTKC,NUTTC,NUT1C,NUT2C,NUT3C,NUT4C,NUT5C,NUT6C/
     *              21,    22,    24,    11,    12,    13,    14,    15,    16/
C  DEFINE BUFFER IN,OUT TAPE UNITS FOR REDUCING CALCULATIONS.
      DATA NUTMR,NUTKR,NUTTR,NUT1R,NUT2R,NUT3R,NUT4R,NUT5R,NUT6R/
     *              21,    22,    24,    11,    12,    13,    14,    15,    16/
C  DEFINE BUFFER IN,OUT TAPE UNITS FOR SUBSTRUCTURE CALCULATIONS.
      DATA NUTMP,NUTKP,NUTTP,NUT1P,NUT2P,NUT3P,NUT4P,NUT5P/
     *              21,    22,    24,    11,    12,    13,    14,    15/
C  DEFINE BUFFER IN,OUT TAPE UNITS FOR PLOTTING.
      DATA NUTPA,NUTFA,NUT1A,NUT2A,NUT3A,NUT4A,NUT5A,NUT6A,NUT7A/
     *              23,    24,    11,    12,    13,    14,    15,    16,    17/
C  DEFINE BUFFER IN,OUT TAPE UNITS FOR SAVEMK.
      DATA NUTMS,NUTKS,NUTTS/
     *              21,    22,    24/
C  DEFINE RESERVE TAPES.
      DATA NRSVT1          / 28     /
C
 1001 FORMAT (12A6)
 1010 FORMAT (10X I5)
C
      IF (NRSVT1 .GT. 0) REWIND NRSVT1
   10 CALL START
      CALL COMENT
      READ (NIT,1001) IFINIT,TAPEID
      IF (IFINIT .EQ. 6HINITIL) CALL INTAPE (NRSVT1,TAPEID)
      IFTRAN = 0
   15 READ (NIT,1001) IOPT
      IF (IOPT   .EQ.6HSTART ) GO TO 10
      IF (IOPT   .EQ.6HGNXYZ ) GO TO 18
      IF (IOPT   .EQ.6HFINEL ) GO TO 20
      IF (IOPT   .EQ.6HXTRAMK) GO TO 30
      IF (IOPT   .EQ.6HMODES ) GO TO 40
      IF (IOPT   .EQ.6HMECHEQ) GO TO 70
      IF (IOPT   .EQ.6HREDUCE) GO TO 80
      IF (IOPT   .EQ.6HSAVEMK) GO TO 90
      IF (IOPT   .EQ.6HSUBSTR) GO TO 100
      IF (IOPT   .EQ.6HPLOT  ) GO TO 110
                                                                  NERROR=1
      GO TO 999
   18 CALL GNXYZ2
      GO TO 15
   20 CALL OFINEL
      GO TO 15
   30 CALL PXTRA
```

```
      GO TO 15
   40 CALL GNIMD
      CALL DYMODE
      GO TO 15
   70 CALL MECHEQ
      GO TO 15
   80 CALL REDUCE
      GO TO 15
   90 CALL SAVE
      GO TO 15
  100 CALL SUBSTR
      GO TO 15
  110 CALL OPLTIO
      GO TO 15
C
  999 CALL ZZBOMB (6HPSLOSH,NERROR)
      END
```

```
@PRT  F1.BASIC
```

```
SIRISHP00000*F1.BASIC
            SUBROUTINE BASIC  (XYZ,JDOF,EUL,KRX,KCX,KRJ,KCJ,KRE,KCE,
     *                         NUTEL,NUTXYZ)
C
C  SUBROUTINE TO READ BASIC FINEL DATA FROM CARD INPUT AND WRITE ON
C  NUTEL AND NUTXYZ.
C  DEVELOPED BY WA BENFIELD,  APRIL 1974.
C
      DIMENSION XYZ(KRX,1), JDOF(KRJ,1), EUL(KRE,1)
      DIMENSION IDATA (14)
C
      DATA NIT,NOT / 5,6 /
 1001 FORMAT (13A6,A2)
      REWIND NUTEL
C  READ JOINT XYZ COORDINATE MATRIX.
   10 CALL READ   (XYZ,NJ,NCX,KRX,KCX)
C  READ JOINT DEGREE OF FREEDOM MATRIX.
      CALL READIM (JDOF,NRJ,NCJ,KRJ,KCJ)
C  READ JOINT EULER ANGLES.
      CALL READ   (EUL,NRE,NCE,KRE,KCE)
C  READ ONE CARD WITH NAMEL FOR FINEL.
C  READ DATA CARDS FOR AXIAL OR BAR OR TRNGL, ETC.
   15 READ  (NIT  ,1001) IDATA
      WRITE (NUTEL,1001) IDATA
      IF (IDATA(1).EQ.6HRETURN) GO TO 100
      GO TO 15
C
  100 REWIND NUTEL
      REWIND NUTXYZ
      WRITE (NUTXYZ) NJ,NCX,NRJ,NCJ,NRE,NCE
      WRITE (NUTXYZ) ((JDOF(I,J),I=1,NRJ),J=1,NCJ)
      WRITE (NUTXYZ) (( XYZ(I,J),I=1, NJ),J=1,NCX)
      WRITE (NUTXYZ) (( EUL(I,J),I=1,NRE),J=1,NCE)
      REWIND NUTXYZ
      RETURN
C
      END
```

```
@PRT  F1.GNIMD
```

```
SIRISHP00000*FI.GNIMD
      SUBROUTINE GNIMD
C
C  ***********************************************************************
C  INPUT DATA READ IN THIS PROGRAM.
C     CALL YREAD  (INITIAL DISPL MODES)
C     RETURN
C
      COMMON /BTAPEM/ NUTM,NUTK,NUTTR,NUTZ,NUT1,NUT2,NUT3,NUT4,NUT5,
     *                NUT6,NUT7
      DIMENSION V(12000), LV(12000)
      DATA KV / 12000 /
      CALL YREAD  (NUTZ, V,LV,KV,NUT1)
      RETURN
      END
```

@PRT FI.GNXYZ2

```
SIRISHP00000*F1.GNXYZ2
      SUBROUTINE GNXYZ2
C
C  MAIN PROGRAM TO GENERATE (XYZ), (JDOF), AND (EUL), AND STORE MATRICES
C  ON UTILITY TAPE.
C  DEVELOPED BY W BENFIELD, C BODLEY, R PHILIPPUS, R WOHLEN.  JULY 1973.
C  LAST REVISION BY R A PHILIPPUS.  JUNE 1974.
C
C  ************************************************************************
C  INPUT DATA READ IN THIS PROGRAM.
C      IOPT                                                FORMAT (A6)
C      IF (IOPT .EQ. 6HBASIC ) CALL READ    (XYZ,NJ,3)
C     *                        CALL READIM  (JDOF,NJ,6)
C     *                        CALL READ    (EUL,NJ,3)
C     *                        CALL FINEL   (SEE SUBRT FOR INPUT)
C      IF (IOPT .EQ. 6HXYZEUL) CALL DATGEN  (SEE SUBRT FOR INPUT)
C      RETURN
C
      COMMON / RWTAPS / NUTEL,NUTXYZ,NUTLT,NUTST,NUTMX,NUTKX,NUTBX
      DATA NIT,NOT / 5,6 /
C
 1001 FORMAT (13A6,A2)
C
      REWIND NUTEL
      READ (NIT,1001) IOPT
      IF (IOPT  .EQ.6HBASIC ) GO TO 10
      IF (IOPT  .EQ.6HFLAT P) GO TO 20
      IF (IOPT  .EQ.6HCNT BM) GO TO 30
      IF (IOPT  .EQ.6HCYLCOR) GO TO 40
      IF (IOPT  .EQ.6HCIRCYL) GO TO 50
      IF (IOPT  .EQ.6HNASTRN) GO TO 60
      IF (IOPT  .EQ.6HSNAP  ) GO TO 70
      IF (IOPT  .EQ.6HXYZEUL) GO TO 80
                                                               NERROR=1
      GO TO 999
C
   10 CALL PBASIC
      REWIND NUTEL
      REWIND NUTXYZ
      RETURN
C
   20                                                          NERROR=2
      GO TO 999
   30                                                          NERROR=3
      GO TO 999
   40                                                          NERROR=4
      GO TO 999
   50                                                          NERROR=5
      GO TO 999
   60                                                          NERROR=6
      GO TO 999
   70                                                          NERROR=7
      GO TO 999
   80 CALL XYZEU2
      REWIND NUTEL
      REWIND NUTXYZ
      RETURN
```

```
      C
  999 CALL ZZBOMB (6HGNXYZ2,NERROR)
      END

@PRT  F1.MECHEQ
```

```
SIRISHP00000*F1.MECHEQ
      SUBROUTINE MECHEQ
      COMMON / BTAPED / NUTKD,NUTLD,NUTDD,NUT11,NUT12,NUT13,NUT14
      DIMENSION V(4000),LV(4000),T(1500,6),TF(600,6),X(405,3),
     *JD(405,6),JVEC(6),E(405,3),R(6,6),RT(6,6),EM(6,6)
      DIMENSION TT(6,600),XR(3),RV(6,6),IVEC(6),ITVC(600)
      DATA KV   ,KR   ,KJ   ,KD   ,KX   , KBT/
     *     4000, 600, 405,    6,    3 , 1500/
      EQUIVALENCE(T,V),(T(4001),LV)
C
C  PROGRAM TO CALCULATE EFFECTIVE MASS FOR SLOSH
C  DEVELOPED BY P . ABBOTT.  DECEMBER 1974.
C
C  FORM RIGID BODY TRANSFORMATION IN GLOBAL
C
 2001 FORMAT (//,10X,13HSLOSH MASS =  ,E11.4,5X,8HCP(X) =   ,E11.4)
C
      CALL READ(X,NR,NC,KJ,KX)
      READ(5,101) N3,NBLAD
  101 FORMAT(16I5)
C
C  N3 IS THE LAST NODE WITH ONLY 3 DOF
C  NBLAD IS THE DOF NO. AT N3.
      DO 6 I=1,N3
      DO 6 J=1,3
    6 JD(I,J)=J+3*(I-1)
      N=N3+1
      DO 7 I=N,NR
      DO 7 J=1,6
    7 JD(I,J)=J+3*N3+(I-N)*6
      CALL WRITIM(JD,NR,6,6HJDOFUL,KJ)
      CALL READ(XR,NF,NC,I,KX)
      CALL READIM(JVEC,NRJ,NCJ,1,KD)
      CALL RBTG1(X,XR,JD,JVEC,T,NR,NRT,NCT,KJ,KBT)
      CALL WRITE(T,NRT,NCT,6HRGLOBL,KBT)
C
C  TRANSFORM TO LOCAL SYSTEM
      CALL READIM(JD,NR,NG,KJ,KD)
      CALL READ(E,NR,NC,KJ,KX)
      DO 100 I=1,6
  100 IVEC(I)=I
      L=1
      M=0
      MI = 0
      DO 10 I=1,NR
      CALL ZERO(JVEC,1,6,1)
      CALL ZERO(R,6,6,KD)
      KROT=3
      IF(I.GT.N3) KROT=6
      K=0
      DO 9 J=1,6
      IF(JD(I,J).LT.0) GO TO 888
      IF(J.LT.4) XR(J)=E(I,J)
      IF(JD(I,J).LE.0) GO TO 9
      IF (I.GT.N3 .AND. JD(I,J).LE.NBLAD) GO TO 9
      K=K+1
      JVEC(J)=K
```

```
      MI = MI+1
      ITVC(MI) = JD(I,J)
    9 CONTINUE
      CALL EULER(XR,R,KD)
      IF(KROT.EQ.6) CALL EULER(XR,R(4,4),KD)
      CALL ZERO   (RV,KROT,K,KD)
      CALL REVADD (1.,R,IVEC,JVEC,RV,KROT,KROT,KROT,K,KD,KD)
      CALL TRANS(RV,RT,KROT,K,KD,KD)
      CALL MULTA(RT,T(L,1),K,KROT,NCT,KD,KBT)
      DO 8 IT=1,K
      DO 8 JT=1,NCT
    8 TF(M+IT,JT)=RT(IT,JT)
      M=M+K
  888 L=L+KROT
   10 CONTINUE
      CALL WRITE(TF,M,NCT,6HROTRAN,KR)
      DO 889 J=1,3
  889 JVEC(J) = J
      CALL ZERO   (T,M,NCT,KBT)
      CALL REVADD (1.,TF,ITVC,JVEC,T,M,NCT,M,NCT,KR,KBT)
      CALL WRITE  (T,M,NCT,6HCORTRN,KBT)
      CALL TRANS(T,TT,M,NCT,KBT,KD)
C
C  READ MASS AND MODES AND DO THE REST
      CALL YREAD  (NUT11,V,LV,KV,NUT14)
      CALL YREAD  (NUT12,V,LV,KV,NUT14)
      CALL YMULT1 (NUT11,NUT12,NUT13,V,LV,KV,NUT14)
      CALL YSTOD  (NUT13,TF,NR,NC,KR,KD,V,LV,KV,NUT14)
      CALL MULT   (TT,TF,R,NCT,NR,NC,KD,KR)
      DO 15 I=1,NCT
      DO 15 J=1,NC
   15 TT(I,J) = R(I,J)
      CALL TRANS(TT,TF,NCT,NC,KD,KR)
      CALL WRITE(TT,NCT,NC,6HTTMPHI,KD)
      CALL PAGEHD
      DO 20  I=1,NC
      CALL MULT(TT(1,I),TF(I,1),EM,NCT,1,NCT,KD,KR)
C  20 CALL WRITE(EM,NCT,NCT,NAME(4HMASS,I),KD)
      CPRESS = EM(2,3)/EM(2,2)
   20 WRITE (6,2001) EM(2,2),CPRESS
      RETURN
      END
```

```
@PRT  F1.OFINEL
```

```
SIRISHP00000*F1.OFINEL
            SUBROUTINE OFINEL
C
C  MAIN PROGRAM TO READ (XYZ), (JDOF), (EUL) AND CALCULATE (ON OPTION)
C  ASSEMBLED FINITE ELEMENT MASS, STIFFNESS, LOAD TRANSFORMATION, AND
C  BUCKLING MATRICES.
C  CALLS FORMA SUBROUTINES FINEL ,YIN   ,YWRITE.
C  DEVELOPED BY W BENFIELD, C BODLEY, R PHILIPPUS, R WOHLEN.  JULY 1973.
C  LAST REVISION BY R A PHILIPPUS.  DECEMBER 1974.
C
            COMMON / RWTAPS / NUTEL,NUTXYZ,NUTLT,NUTST,NUTMX,NUTKX,NUTBX
            COMMON / BTAPEF / NUTM,NUTK,NUT1,NUT2,NUT3
            COMMON / DUMMY / V(12000)
            COMMON / IDUMMY / LV(12000)
            COMMON / RESTAP / NRSVT1
            DIMENSION XYZ(2000,3), JDOF(2000,6), EUL(2000,3)
            EQUIVALENCE (XYZ,V), (EUL,V(6001)), (JDOF,LV)
            DATA  KRX, KCX,  KRJ, KCJ,  KRE, KCE,   KV /
           *      2000,    3, 2000,    6, 2000,    3,12000 /
C  READ XYZ,JDOF,EUL FROM NUTXYZ CREATED IN PROGRAM DATGEN.
            REWIND NUTXYZ
            READ (NUTXYZ) NJ,NCX,NRJ,NCJ,NRE,NCE
                                                                  NERROR=1
            IF (NCX .NE. 3) GO TO 999
                                                                  NERROR=2
            IF (NRJ .NE. NJ .OR. NCJ .NE. 6) GO TO 999
                                                                  NERROR=3
            IF (NRE .NE. NJ .OR. NCE .NE. 3) GO TO 999
                                                                  NERROR=4
            IF ( NJ.GT.KRX .OR. NCX.GT.KCX .OR.
           *     NRJ.GT.KRJ .OR. NCJ.GT.KCJ .OR.
           *     NRE.GT.KRE .OR. NCE.GT.KCE) GO TO 999
            READ (NUTXYZ) ((JDOF(I,J),I=1,NRJ),J=1,NCJ)
            READ (NUTXYZ) (( XYZ(I,J),I=1, NJ),J=1,NCX)
            READ (NUTXYZ) (( EUL(I,J),I=1,NRE),J=1,NCE)
            CALL WTAPE  ( XYZ,NJ ,NCX,6HXYZ   ,KRX,NRSVT1)
            CALL WTAPE  (JDOF,NRJ,NCJ,6HJDOF  ,KRJ,NRSVT1)
            CALL WTAPE  ( EUL,NRE,NCE,6HEUL   ,KRE,NRSVT1)
            CALL FINEL  (XYZ,JDOF,EUL,NUTEL,NJ,
           *             NUTM,NUTK,NUTLT,NUTST,NUTBX,
           *             V,LV,KV,KRX,KRJ,KRE,
           *             NUTMX,NUTKX,NUT1,NUT2,NUT3)
            CALL YWRITE (NUTM,4HMASS  ,V,LV,KV)
            CALL YWTAPE (NUTM,6HMASS  ,V,LV,KV,NRSVT1)
            CALL YWRITE (NUTK,4HSTIF  ,V,LV,KV)
            CALL LTAPE  (NRSVT1)
            RETURN
C
        999 CALL ZZBOMB (6HOFINEL,NERROR)
            END
```

WPRT F1.OPLTIO

```
SIRISHP00000*F1.OPLT10
      SUBROUTINE OPLT10
      COMMON / BTAPED / NUTKD,NUTLD,NUTOD,NUT1,NUTP,NUT3,NUT4
      DIMENSION XYT(45,3), PLOC(90,20), XP(200), YP(200), XPYP(90,5),
     *          DX(45), DY(45), DXL(45), DYL(45), PTITLE(4),
     *          SCALEM(20), FREQM(20), BIGP(600,6),IVEC(45),JDOF(600,6),
     *          V(4000), LV(4000)
      DATA KJ, KP, KM, KPLOT,KBIGP,KBIGM,    KV/
     *     45, 90, 20,   200,  600,    6, 4000/
      DATA EPS/1.E-15/
      DATA NIT,NOT/5,6/
 1001 FORMAT (10X I5,I5)
 2001 FORMAT (10X, F10.4,F10.4)
 2002 FORMAT (//10X, 9HMODE NO. I3, //)
 2003 FORMAT (//10X, 9HMODE NO. I3, 5X 7HSCALE = F6.3)
 5001 FORMAT (5HMODE=I3, 1X 5HFREQ=F7.4, 1X 5HSHFT=I4, 1X 4HSCL=F4.3)
      CALL IDFRMV (12H G I BULTMAN ,12H BIN S 190    ,6H5 6325 ,
     *              12HHARDCOPY     )
C
C  READ IVEC. IN SURFACE JOINT ORDER FROM CENTER OUT. VALUE IS ROW
C  NUMBER IN XYZ,JDOF,EULER.
      CALL READIM (IVEC, I1,NJS, 1,KJ)
C  READ XYZ.
      CALL READ   (BIGP, NJF,I3, KBIGP,3)
      DO 5 I=1,NJS
      JNT = IVEC(I)
      XYT(I,1) = BIGP(JNT,1)
    5 XYT(I,2) = BIGP(JNT,2)
C  READ EULER.
      CALL READ   (BIGP, NJF,I3, KBIGP,3)
      DO 6 I=1,NJS
      JNT = IVEC(I)
    6 XYT(I,3) = BIGP(JNT,3)
      CALL WRITE  (XYT, NJS,3, 3HXYT, KJ)
      IMS = 1
      ISHIFT = 0
      CALL READ   (FREQM, NM,I1,KM,1)
C  READ MODES. FIND LARGEST VALUE (ABS) IN EACH MODE. (SCALEM).
      CALL YREAD  (NUTP,V,LV,KV,NUT1)
      CALL YSTOD  (NUTP,BIGP,NBIGP,NM,KBIGP,KBIGM,V,LV,KV,NUT1)
      DO 9 J=1,NM
      BIGM = 0.0
      DO 8 I=1,NBIGP
      VALM = ABS(BIGP(I,J))
      IF (VALM .GT. BIGM) BIGM=VALM
    8 CONTINUE
    9 SCALEM(J) = 1./BIGM
      CALL WRITE  (SCALEM, 1,NM, 6HSCALEM, 1)
C  READ JDOF.
      CALL READIM (JDOF, NJF,I6, KBIGP,6)
C  COMPRESS BIGP INTO PLOC. ORDER IS U,V IN JOINT ORDER GIVEN BY IVEC.
      DO 12 I=1,NJS
      JNT = IVEC(I)
      IU = JDOF(JNT,1)
      IV = JDOF(JNT,2)
      IPU = 2*I-1
      IPV = 2*I
```

```
          DO 12 J=1,NM
          PLOC(IPU,J) = 0.0
          IF (IU .GT. 0) PLOC(IPU,J)=BIGP(IU,J)
          PLOC(IPV,J) = 0.0
          IF (IV .GT. 0) PLOC(IPV,J)=BIGP(IV,J)
       12 CONTINUE
          CALL WRITE  (PLOC, 2*NJS,NM, 4HPLOC, KP)
C
C  DEFINE UNDISTURBED SHAPE, XP,YP IN PLOT X,Y. LEFT,RIGHT.
          NJ = NJS
          NJM1 = NJ-1
          IJ = NJ+1
          DO 10 IP=1,NJM1
          IJ = IJ-1
          XP(IP) = -XYT(IJ,2)
       10 YP(IP) =  XYT(IJ,1)
          DO 15 IJ=1,NJ
          IP = NJM1+IJ
          XP(IP) = XYT(IJ,2)
       15 YP(IP) = XYT(IJ,1)
C
          DO 199 IM=1,NM
          MODENO = IMS-1+IM
          SCALE  = SCALEM(IM)
          FREQ = FREQM(IM)
C  CALCULATE DELTA-X, DELTA-Y. IN JAG X,Y. LEFT=DXL,DYL. RIGHT.
          CALL PAGEHD
          WRITE (NOT,2002) MODENO
          DO 50 IJ=1,NJ
          IXL = 2*IJ-1
          IYL = 2*IJ
          ANG = XYT(IJ,3)/57.29577951
          CJ = COS(ANG)
          SJ = SIN(ANG)
          DXJAG = PLOC(IXL,IM)*CJ - PLOC(IYL,IM)*SJ
          DYJAG = PLOC(IXL,IM)*SJ + PLOC(IYL,IM)*CJ
          WRITE (NOT,2001) DYJAG,DXJAG
          DX(IJ) = XYT(IJ,1) + SCALE*DXJAG
          DXL(IJ)=+XYT(IJ,1) - SCALE*DXJAG
          DY(IJ) = XYT(IJ,2) + SCALE*DYJAG
       50 DYL(IJ)=-XYT(IJ,2) + SCALE*DYJAG
C  PACK DY,DX INTO XP,YP. LEFT,RIGHT.
          IP1 = NJM1+NJ+1
          IP2 = IP1-1+NJM1
          IJ = NJ+1
          DO 60 IP=IP1,IP2
          IJ = IJ-1
          XP(IP) = DYL(IJ)
       60 YP(IP) = DXL(IJ)
          IP1 = NJM1+NJ+NJM1
          DO 65 IJ=1,NJ
          IP = IP1+IJ
          XP(IP) = DY(IJ)
       65 YP(IP) = DX(IJ)
C  PACK XP,YP FOR PRINTING.
          NJ2M1 = 2*NJ-1
          DO 80 I=1,NJ2M1
```

```
            XPYP(I,1) = XP(I)
            XPYP(I,2) = YP(I)
            XPYP(I,3) = 0.0
            XPYP(I,4) = XP(I+NJ2M1)
         80 XPYP(I,5) = YP(I+NJ2M1)
            CALL WRITE  (XPYP, NJ2M1,5, 4HXPYP, KP)
            WRITE (NOT,2003) MODENO,SCALE
      C  PUT IN TANK TOP,BOT AT CENTERLINE.
            XP(IP+1) = 0.
            YP(IP+1) =16.
            XP(IP+2) = 0.
            YP(IP+2) = 0.
            NRP = IP+2
            IFSAME = 1
            IFCURV = 1
            IFLIFT = 1
            ENCODE (40,5001,PTITLE) MODENO,FREQ,ISHIFT,SCALE
            CALL PLOT1  (XP,YP,NRP,1, -10.,2., 6HY-TANK,6HX-TANK,PTITLE,
           *                  IFSAME,IFCURV,IFLIFT,KPLOT)
        199 CONTINUE
      C
            CALL PLTND
            RETURN
            END


@PRT  F1.OYMODE
```

```
SIRISHP00000*F1.OYMODE
      SUBROUTINE OYMODE
C
C     MAIN PROGRAM TO TEST ITERATIVE RAYLEIGH-RITZ METHOD OF DR. JOHN ADMIRE
C     TECHNIQUE = COMPOSITE STRUCTURE.
C     VERSION = NON-SWEEPING.
C     PROGRAMMING LOGIC = SPARSE.
C     MAXIMUM SIZE OF MASS,STIF = 1820.
C     MAXIMUM NU = 70
C     DEVELOPED BY R L WOHLEN AND R A PHILIPPUS.  MARCH 1972.
C     LAST DIMENSION CHANGE (FOR 150000) BY HRUDA  02APR74.
C     LAST REVISION BY R A PHILIPPUS.  JULY 1974.
C
C     ******************************************************************
C     INPUT DATA READ IN THIS PROGRAM.
C        NW                                                 FORMAT (10X,I5)
C        NU                                                 FORMAT (10X,I5)
C        SHIFT                                              FORMAT (10X,E10)
C        MAXIT                                              FORMAT (10X,I5)
C        IPUNCH                                             FORMAT (A6)
C        RETURN
C
C     DEFINITION OF INPUT VARIABLES.
C     NW     = NUMBER OF MODES WANTED.
C     NU     = NUMBER OF RAYLEIGH-RITZ MODES TO USE.
C     SHIFT  = SHIFT VALUE TO USE.
C     MAXIT  = MAXIMUM NUMBER OF ITERATIONS TO BE PERFORMED.
C     IPUNCH = PUNCH CARD OPTION FOR W2, FREQ, MODES, AND TMODES.
C            = 6HPUNCH , FOR PUNCH CARD OUTPUT.
C            = 6HNOPUNC, NO PUNCH CARD OUTPUT.
C
      COMMON /BTAPEM/ NUTM,NUTK,NUTTR,NUTZ,NUTF,NUT1,NUT2,NUT3,NUT4,
     *                NUT5,NUT6,NUT7
      COMMON / RESTAP / NRSVT1
      COMMON / RTRANS / IFTRAN
C
      DIMENSION V(10920), LV(10920), W2(70), W(70), FREQ(70),
     *          A( 70, 70), S( 70, 70), MH(10)
C
      EQUIVALENCE (V(3641),S), (LV(3641),A)
C
      DATA NIT,NOT / 5,6 /
      DATA    KV, KA /
     *     10920, 70 /
      DATA NITER1, NITER2,   TOLZ,  TOLW2/
     *          0,      1, 1.E-06, 1.E-04/
      DATA IFPRNT/1000/
 1001 FORMAT (10X, 4I5)
 1010 FORMAT (10X, E10.0)
 1020 FORMAT (12A6)
C
      READ (NIT,1001) NW
      READ (NIT,1001) NU
      READ (NIT,1010) SHIFT
      READ (NIT,1001) MAXIT
      READ (NIT,1020) IPUNCH
C
```

```
      CALL YMODE2 (NUTM,NUTK,NUTZ,W2,W,FREQ,NW, V,LV,A,S, KV,KA,
     *                 NUT1,NUT2,NUT3,NUT4,NUT5,NUT6,NUT7,
     *                 IFPRNT,MAXIT,
     *                 NU,NITER1,NITER2,SHIFT,TOLZ,TOLW2)
C
      CALL WRITE  (W2,NW,1,2HW2,KA)
      CALL WRITE  (W ,NW,1,1HW ,KA)
      CALL WRITE  (FREQ,NW,1,4HFREQ,KA)
      CALL YWRITE (NUTZ,5HMODES,V,LV,KV)
      IF (NRSVT1 .GT. 0) CALL WTAPE  (W2,NW,1,2HW2,KA,NRSVT1)
      IF (NRSVT1 .GT. 0) CALL WTAPE  (FREQ,NW,1,4HFREQ,KA,NRSVT1)
      IF (NRSVT1 .GT. 0) CALL YWTAPE (NUTZ,5HMODES,V,LV,KV,NRSVT1)
      CALL YDTOS  (FREQ,NUTF,NW,1,KA,1,V,LV,KV,NUT7)
      IF (IFTRAN .EQ. 0) GO TO 100
      REWIND NUTTR
      CALL YINI   (NUTTR,MH,1,10)
      NRT = MH(1)
      NCT = MH(2)
      CALL YDISA  (NUTZ,1,1,NUT1,NCT,NW,V,LV,KV,NUT7)
      CALL YMULT  (NUTTR,NUT1,NUT2,V,LV,KV,NUT7)
      REWIND NUTZ
      CALL YINI   (NUTZ,NRM,1,1)
      NRTM = NRM - NCT + NRT
      CALL YZERO  (NUT1,NRTM,NW)
      CALL YASSEM (NUT2,1,1,NUT1,V,LV,KV,NUT5,NUT6,NUT7)
      NCTP1 = NCT + 1
      NRX = NRM - NCT
      CALL YDISA  (NUTZ,NCTP1,1,NUT3,NRX,NW,V,LV,KV,NUT7)
      NRTP1 = NRT + 1
      CALL YASSEM (NUT3,NRTP1,1,NUT1,V,LV,KV,NUT5,NUT6,NUT7)
      CALL YWRITE (NUT1,6HTMODES,V,LV,KV)
      IF (NRSVT1 .GT. 0) CALL YWTAPE (NUT1,6HTMODES,V,LV,KV)
  100 IF (NRSVT1 .GT. 0) CALL LTAPE (NRSVT1)
      IF (IPUNCH .NE. 5HPUNCH ) RETURN
      CALL PUNCH  (FREQ,NW,1,4HFREQ,KA)
      CALL YPUNCH (NUTZ,5HMODES,V,LV,KV)
      IF (IFTRAN .NE. 0) CALL YPUNCH (NUT1,6HTMODES,V,LV,KV)
      RETURN
C
      END
```

@PRT F1.PBASIC

```
SIRISHP00000*F1.PBASIC
               SUBROUTINE PBASIC
      C  PROGRAM TO CALL BASIC.
               COMMON / RWTAPS / NUTEL,NUTXYZ,NUTLT,NUTST,NUTMX,NUTKX,NUTBX
               DIMENSION XYZ(2000,3), JDOF(2000,6), EUL(2000,3)
               DATA KRX, KCX,  KRJ, KCJ,  KRE, KCE /
              *    2000,    3, 2000,    6, 2000,    3 /
      C
               CALL BASIC  (XYZ,JDOF,EUL,KRX,KCX,KRJ,KCJ,KRE,KCE,
              *                NUTEL,NUTXYZ)
               RETURN
               END


@PRT  F1.PXTRA
```

```
SIRISHP00000*F1.PXTRA
          SUBROUTINE PXTRA
C   OVERLAY PROGRAM TO REVADD EXTRA MASS AND STIFFNESS MATRICES TO
C   EXISTING MASS AND STIFFNESS MATRICES.
C
C   ***********************************************************************
C   INPUT DATA READ IN THIS PROGRAM.
C       CALL READIM (IJVEC,1,NCI)
C       CALL YREAD  (MASS MATRIX)
C       CALL YREAD  (STIF MATRIX)
C       RETURN
C
        COMMON / BTAPET / NUTM,NUTK,NUT1,NUT2,NUT3,NUT4,NUTX
        DIMENSION V(12000), LV(12000), IJVEC(2000)
        DATA    KV,  KIV /
      *     12000, 2000 /
C   READ IJVEC FOR EXTRA M,K MATRICES.
        CALL READIM (IJVEC,NRI,NCI,1,KIV)
        DO 10 I=1,NCI
        IF (IJVEC(I) .NE. 0) GO TO 20
     10 CONTINUE
        DO 15 I=1,NCI
     15 IJVEC(I) = I
C   READ EXTRA MASS MATRIX.
     20 CALL YREAD  (NUTX,V,LV,KV,NUT1)
        CALL YREVAD (1.,NUTX,IJVEC,IJVEC,NUTM,V,LV,KV,NUT1,NUT2,NUT3,NUT4)
C   READ EXTRA STIFFNESS MATRIX.
        CALL YREAD  (NUTX,V,LV,KV,NUT1)
        CALL YREVAD (1.,NUTX,IJVEC,IJVEC,NUTK,V,LV,KV,NUT1,NUT2,NUT3,NUT4)
C       CALL YWRITE (NUTM,5HMXTRA ,V,LV,KV)
C       CALL YWRITE (NUTK,5HKXTRA ,V,LV,KV)
        RETURN
        END


@PRT  F1.REDUCE
```

```
SIRISHP00000*F1.REDUCE
            SUBROUTINE REDUCE
      C
      C  OVERLAY PROGRAM TO REDUCE STIFFNESS AND MASS MATRIX.
      C  DOF TO BE REDUCED MUST BE POSITIONED FIRST IN MATRIX.
      C  REDUCING TRANSFORMATION IS STORED ON NUTTR.
      C  DEVELOPED BY WA BENFIELD, FEBRUARY 1974.
      C
      C  *****************************************************************
      C  INPUT  DATA READ IN THIS PROGRAM.
      C      NR                                          FORMAT (10X,I5)
      C      CALL READIM (IJVEC,1,NCI)
      C      RETURN
      C
      C  DEFINITION OF INPUT VARIABLES.
      C  NR      = NUMBER OF ROW-COLS IN REDUCED MATRIX.
      C  IJVEC   = IJVEC TO REARRANGE ROWS AND COLS BEFORE REDUCING.
      C
            COMMON /BTAPER / NUTMR,NUTKR,NUTTR,NUTR1,NUTR2,NUTR3,NUTR4,NUTR5,
           *                 NUTR6
            COMMON / RTRANS / IFTRAN
      C
            DIMENSION V(10000), LV(10000)
            DIMENSION IV1(3000), IV2(3000)
      C
            DATA NIT, NOT / 5, 6 /
            DATA KV  / 10000 /
            DATA KIV /  3000 /
      C
       1001 FORMAT  (6(10X,I5))
       2001 FORMAT (/// 20X14HOVERLAY REDUCE,
           *         //  10X42HNUMBER OF ROWS AND COLS BEFORE REDUCING = ,I5,
           *          /   10X42HNUMBER OF ROWS AND COLS AFTER REDUCING = ,I5)
            DO 10 I=1,KIV
         10 IV2(I) = I
      C
      C  READ NUMBER OF ROWS IN REDUCED MATRIX.
            READ (NIT,1001) NR
            REWIND NUTKR
            CALL YINI (NUTKR,NRK,1,1)
            CALL PAGEHD
            WRITE (NOT,2001) NRK, NR
      C
      C  READ REARRANGING IVEC.
            CALL READIM (IV1,NR1,NCI,1,KIV)
            DO 12 I=1,NCI
            IF (IV1(I) .NE. 0) GO TO 17
         12 CONTINUE
            DO 15 I=1,NCI
         15 IV1(I) = I
      C
      C  REARRANGE MASS, STIFF, AND TRANS (IF ANY) MATRICES.
         17 CALL YZERO  (NUTR1,NRK,NRK)
            CALL YZERO  (NUTR2,NRK,NRK)
            CALL YREVAD (1.0,NUTMR,IV1,IV1,NUTR1,V,LV,KV,NUTR2,NUTR3,NUTR4,
           *             NUTR5)
            CALL YREVAD (1.0,NUTKR,IV1,IV1,NUTR2,V,LV,KV,NUTR3,NUTR4,NUTR5,
```

```
     *                NUTMR)
      IF (IFTRAN .EQ. 0) GO TO 20
      REWIND NUTTR
      CALL YINI    (NUTTR,LV,1,2)
      CALL YZERO   (NUTR3,LV(1),LV(2))
      CALL YREVAD  (1.0,NUTTR,IV2,IV1,NUTR3,V,LV,KV,NUTR4,NUTR5,NUTMR,
     *                NUTKR)
   20 CALL YSHED2  (NUTR2,NUTKR,NUTTR,NR,1,V,LV,KV,NUTR4,NUTR5,NUTR6,
     *                NUTMR)
      CALL YWRITE  (NUTTR,2HTR      ,V,LV,KV)
      CALL YWRITE  (NUTKR,2HKR      ,V,LV,KV)
      IF (IFTRAN .EQ. 0) GO TO 30
      CALL YMULTB  (NUTR3,NUTTR,V,LV,KV,NUTR4,NUTR5)
      CALL YWRITE  (NUTTR,6HMULTTR,V,LV,KV)
   30 CALL YBTAB   (NUTR1,NUTTR,NUTMR,V,LV,KV,NUTR4,NUTR5)
      CALL YWRITE   (NUTMR,2HMR      ,V,LV,KV)
      IFTRAN = 1
      RETURN
      END
```

@PRT F1.SAVE

```
SIRISHP00000*F1.SAVE
            SUBROUTINE SAVE
      C
      C  OVERLAY PROGRAM TO SAVE MASS, STIFFNESS, AND REDUCING TRANSFORMATION
      C  MATRICES ON FORMA LIBRARY TAPE.
      C  DEVELOPED BY WA BENFIELD, FEBRUARY 1974.
      C
      C  ***********************************************************************
      C  INPUT DATA READ IN THIS PROGRAM.
      C     NAMEM,NAMEK,NAMET                               FORMAT (3A6)
      C     RETURN
      C
      C  DEFINITION OF INPUT VARIABLES.
      C  NAMEM  = NAME OF MASS MATRIX TO USE ON FORMA TAPE.
      C         = 6H       , MASS IS NOT WRITTEN ON FORMA TAPE.
      C  NAMEK  = NAME OF STIF MATRIX TO USE ON FORMA TAPE.
      C         = 6H       , STIF IS NOT WRITTEN ON FORMA TAPE.
      C  NAMET  = NAME OF TRANSFORMATION MATRIX TO USE ON FORMA TAPE.
      C         = 6H       , TRANSFORMATION IS NOT WRITTEN ON FORMA TAPE.
      C
            COMMON / BTAPES / NUTMS,NUTKS,NUTTS
            COMMON / RESTAP / NRSVT1
            COMMON / RTRANS / IFTRAN
      C
            DIMENSION V(15000), LV(15000)
      C
            DATA KV / 15000 /
            DATA NIT, NOT / 5, 6 /
      C
       1001 FORMAT (12A6)
      C
            READ (NIT,1001) NAMEM,NAMEK,NAMET
            IF (NAMEM .NE. 6H      ) CALL YWTAPE (NUTMS,NAMEM,V,LV,KV,NRSVT1)
            IF (NAMEK .NE. 6H      ) CALL YWTAPE (NUTKS,NAMEK,V,LV,KV,NRSVT1)
            IF (IFTRAN .NE. 0 .AND. NAMET .NE. 6H      ) CALL YWTAPE (NUTTS,
           *                                         NAMET,V,LV,KV,NRSVT1)
            CALL LTAPE  (NRSVT1)
            RETURN
            END

WPRT  F1.SUBSTR
```

```
SIRISHP00000*F1.SUBSTR
      SUBROUTINE SUBSTR
C
C OVERLAY PROGRAM TO COMBINE SUBSTRUCTURE MASS AND STIFFNESS MATRICES
C TOGETHER.
C DEVELOPED BY WA BENFIELD. FEBRUARY 1974.
C
C  ******************************************************************.**
C  INPUT DATA READ IN THIS PROGRAM.
C     NSUBS                                              FORMAT (10X,I5)
C     NR                                                 FORMAT (10X,I5)
C     NTRAN                                              FORMAT (A6)
C     DO 30 K=1,NSUBS
C     NSRC                                               FORMAT (10X,I5)
C     CALL YREAD  (MASS MATRIX)
C  30 CALL YREAD  (STIF MATRIX)
C     IF (NTRAN .EQ. 6HTRANS ) CALL YREAD (TRANS MATRIX)
C     RETURN
C
C  DEFINITION OF INPUT VARIABLES.
C  NSUBS  = NUMBER OF SUBSTRUCTURES TO BE READ IN.
C  NR     = NUMBER OF ROW-COLS IN TOTAL MASS-STIF MATRICES.
C  NTRAN  = OPTION TO READ IN TRANSFORMATION MATRIX.
C         = 6HTRANS , TRANSFORMATION MATRIX IS READ IN.
C         = 6HNOTRAN, TRANSFORMATION MATRIX IS NOT READ IN.
C  NSRC   = START ROW-COL TO ASSEMBLE SUBSTRUCTURE INTO.
C
      COMMON / BTAPEP / NUTMP,NUTKP,NUTTP,NUT1P,NUT2P,NUT3P,NUT4P,NUT5P
      COMMON / RTRANS / IFTRAN
C
      DIMENSION V(10000), LV(10000)
      DIMENSION IV1(3000)
C
      DATA NIT, NOT / 5, 6 /
      DATA KV  / 10000 /
      DATA KIV /  3000 /
C
 1001 FORMAT (6(10X,I5))
 1002 FORMAT (12A6)
C
C  READ NUMBER OF SUBSTRUCTURES TO COMBINE TOGETHER.
      READ (NIT,1001) NSUBS
C
C  READ NUMBER OF ROW-COLS IN FINAL MATRIX, AND IF TRANSFORMATION WILL
C  BE INPUT.
      READ (NIT,1001) NR
      READ (NIT,1002) NTRAN
      CALL YZERO  (NUTMP,NR,NR)
      CALL YZERO  (NUTKP,NR,NR)
      DO 30 K=1,NSUBS
C
C  READ STARTING ROW-COL NUMBER TO ASSEMBLE SUBSTRUCTURE INTO.
      READ (NIT,1001) NSRC
      IF (NSUBS .EQ. 1 .AND. NSRC .EQ. 1) GO TO 40
      DO 10 I=1,NR
   10 IV1(I) = NSRC + I - 1
C
```

```
      C   READ SUBSTRUCTURE MASS MATRIX.
              CALL YREAD   (NUT1P,V,LV,KV,NUT2P)
              CALL YREVAD (1.0,NUT1P,IV1,IV1,NUTMP,V,LV,KV,NUT2P,NUT3P,NUT4P,
             .                 NUT5P)
      C
      C   READ SUBSTRUCTURE STIFF MATRIX.
              CALL YREAD   (NUT1P,V,LV,KV,NUT2P)
              CALL YREVAD (1.0,NUT1P,IV1,IV1,NUTKP,V,LV,KV,NUT2P,NUT3P,NUT4P,
             .                 NUT5P)
           30 CONTINUE
      CCCCC CALL YWRITE (NUTKP,6HTOTSTF,V,LV,KV)
      CCCCC CALL YWRITE (NUTMP,6HTOTMAS,V,LV,KV)
              GO TO 50
           40 CALL YREAD   (NUTMP,V,LV,KV,NUT2P)
              CALL YREAD   (NUTKP,V,LV,KV,NUT2P)
      C
      C   READ TRANSFORMATION MATRIX (IF ANY).
           50 IF (NTRAN .NE. 6HTRANS ) RETURN
              CALL YREAD   (NUTTP,V,LV,KV,NUT2P)
              IFTRAN = 1
              RETURN
              END
```

@PRT   F1.XYZEU2

```
SIRISHP00000*F1.XYZEU2
            SUBROUTINE XYZEU2
      C
      C     SUBROUTINE XYZJAG (NUTEL,NUTXYZ)
      C
            DIMENSION XYZ(1200,3),EUL(1200,3),JDOF(1200,6)
            DIMENSION ICONF(1500,8),ICONS(200,4),ICONB(200,4)
            DIMENSION XYZB(200,3),EULB(200,3)
            DIMENSION RB(20),AI(20),XO(20),NP INTS(20),NELEMI(20),XOI(20)
            DIMENSION AN5(10),AN6(11),INTG(2)
      C
            COMMON / RWTAPS / NUTEL,NUTXYZ,NUTLT,NUTST,NUTMX,NUTKX,NUTBX
      C
            DATA KX,KS,KB/1200,200,20/
            DATA PENTA,HEXA/5HPENTA,4HHEXA/
            DATA NCX,NCJ/3,6/
            DATA AN6/6HRETURN,6HFLUID ,6HGRAVTY,6H      ,6HTRNGL ,6HQUAD  ,
     *         6HM1    ,6HM2    ,6HK1    ,6HK2    ,6HCNST  /,
     *       AN5/5HRO   ,5HBLKM ,5HGVX  ,5HGVY  ,5HGVZ  ,5HE    ,5HNU   ,
     *         5HTMAS ,5HTMEM ,5HTBEN  /,
     *      INTG/0,0/
            DATA ZZERO/0.0/
            DATA EPS/1.0E-05/
      C
      C
       1001 FORMAT (16I5)
       1002 FORMAT (8F10.0)
       1003 FORMAT (8A10)
       3001 FORMAT (A6)
       3002 FORMAT (5(A6,4X))
       3003 FORMAT (3(A5,E10.3))
       3004 FORMAT (4I5,3E10.3)
       3005 FORMAT (5I5,3E10.3)
       3006 FORMAT (9I5)
       5001 FORMAT (10(/)
     *      15X,  22HRADIUS OF THE SPHERE =,  F10.3,//
     *      15X,  22H  FLUID MASS DENSITY =,  F10.7,//
     *      15X,  22H        BULK MODULUS =,  F14.3,//
     *      15X,  22H                 GVX =,  F10.3,//
     *      15X,  22H                 GVY =,  F10.3,//
     *      15X,  22H                 GVZ =,  F10.3,//
     *      15X,  22HBLADDER MASS DENSITY =,  F10.7,//
     *      15X,  22H                   E =,  F14.3,//
     *      15X,  22H                  NU =,  F10.7,//
     *      15X,  22H   BLADDER THICKNESS =,  F10.7)
       5002 FORMAT (10(/)
     *      15X,  22H                NCIR =, I5,//
     *      15X,  22H               NRBAY =, I5)
      C
      C     ------------------------------------------------------------------
      C
      C     I N P U T S ----------------
      C
      C     READ (5,1002) R                                   (8F10.0)
      C     READ (5,1002) RHOF,BLKM,GVX,GVY,GVZ               (8F10.0)
      C     READ (5,1002) RHOBLD,EBLD,ANUBLD,TBLD             (8F10.0)
      C     READ (5,1001) NCIR,NRBAY                          (16I5)
```

```
C       CALL READ (AI,NRA,NCA,1,KB)                             (5 OF THEM)
C       CALL READ (RB,NRR,NCR,1,KB)                             (1 X NRBAY)
C       CALL READIM (NPINTS, NRN,NCN,1,KB)                      (1 X (NRBAY+1))
C       CALL READIM (NELEMI,NRE,NCE,1,KB)                       (1 X NRBAY)
C       READ (5,1002) (XO(I) ,I=1,NP1)                          (8F10.0)
C       READ (5,1002) (XOI(I),I=1,NP2)                          (8F10.0)
C       DO 200 II=2,NRBAY
C       READ (5,1002) (XOI(K),K=1,NP1)                          (8F10.0)
C 200   CONTINUE
C       READ (5,3002) ELMID1,ELMID2                             (5(A6,4X))
C       READ (5,3002) ELMID1,ELMID2                             (5(A6,4X))
C       DO 500 II=3,NRBAY1
C       READ (5,3002) ELMID1,ELMID2                             (5(A6,4X))
C 500   CONTINUE
C
C       ----------------------------------------------------------------
C
C       E X P L A N A T I O N S -----------
C
C-             AI=COEFFICIENTS OF THE FREE SURFACE POLYNOMIAL.
C-              R=RADIUS OF THE SPHERE.
C-           NCIR=NO.OF CIRCUMFERENTIAL PTS. AROUND THE MODEL.
C-          NRBAY=NO.RADIAL BAYS IN THE MODEL.
C-             RB=RADIAL BAY RADII
C-         THETAT=TOTAL MODEL ANGLE (=180.0)
C-        NPOINTS=NO.OF POINTS ON EACH VERTICAL LINE IN THE PLANE.(VECTOR)
C-             XO=XO-S FOR FOR THE CENTERLINE.
C-            XOI=X-VALUES IN EACH VERTICAL LINE IN THE PLANE (NO.=NPOINTS)
C-                 (NOTE&POINTS ON SPHERE AND BLADDER ARE REPLACED BY ACTUAL
C-                  CALCULATED VALVES). STARTING FROM IN EXCEPTING CENT.LINE.
C-         ELMID1=FIRST ELEMENT(FROM BOTTOM) FOR THE ANNULAR SPACE ID.
C-         ELMID2=LAST  ELEMENT(FROM BOTTOM) FOR THE ANNULAR SPACE ID.
C           PENTA=PENTAHEDRON
C            HEXA=HEXAHEDRON
C-         NELEMI=NO.OF ELEMENTS IN EACH BAY (PLANER MODEL) (VECTOR)
C
C
C-      BLADDER IS SUPPOSED TO BE ATTACHED AT DIAMETRAL PLANE
C
C
C-           RHOF=MASS DENSITY OF THE FLUID
C-           BLKM=BULK MODULUS FOR THE FLUID
C-            GVX=GRAVITY ACCELERATION IN X-DIRECTION
C-            GVY=GRAVITY ACCELERATION IN Y-DIRECTION
C-            GVZ=GRAVITY ACCELERATION IN Z-DIRECTION
C-         RHOBLD=MASS DENSITY OF THE BLADDER
C-           EBLD=YOUNG-S MODULUS FOR THE BLADDER.
C-         ANUBLD=POISSON-S RATIO FOR THE BLADDER
C-           TBLD=THICKNESS OF THE BLADDER
C
C-NOTE-THE TOP ELEMENTS IN THE FIRST ANNULAR SPACE HAVE TO BE TETRAS.
C
C       ----------------------------------------------------------------
C
        REWIND NUTEL
        REWIND NUTXYZ
C
```

```
      PI = ATAN2(0.0,-1.0)
      RADDEG = ATAN2(1.0,1.0) / 45.0
C
C-    INITIALIZE SOME VARIABLES REQUIRED
C
      NDP  = 0
      ND   = 0
      NEL  = 0
      NELS= 0
      NELB= 0
      NDPB= 0
C
      CALL ZERO (XYZ,KX,3,KX)
      CALL ZERO (XYZB,KS,3,KS)
      CALL ZERO (EUL,KX,3,KX)
      CALL ZERO (EULB,KS,3,KS)
      DO 20 I=1,1500
      DO 20 J=1,8
   20 ICONF(I,J) = 0
      DO 30 I=1,KS
      DO 30 J=1,8
      ICONS(I,J) = 0
   30 ICONB(I,J) = 0
C
C-    READ IN DATA
C
      READ (5,1002) R
      READ (5,1002) RHOF,BLKM,GVX,GVY,GVZ
      READ (5,1002) RHOBLD,EBLD,ANUBLD,TBLD
C
      READ (5,1001) NCIR,NRBAY
C
      WRITE (6,5001) R,RHOF,BLKM,GVX,GVY,GVZ,RHOBLD,EBLD,ANUBLD,TBLD
      WRITE (6,5002) NCIR,NRBAY
C
      CALL READ (AI,NRA,NCA,1,KB)
      CALL READ (RB,NRR,NCR,1,KB)
      CALL READIM (NPINTS, NRN,NCN,1,KB)
      CALL READIM (NELEMI,NRE,NCE,1,KB)
C
      NCIR1 = NCIR - 1
      ANCIR = NCIR
      ANCIR1 = NCIR1
      THETAT = PI
      THETAD = THETAT / ANCIR1
      NDV2 = NCIR1
      NQ = NCIR1 / 2
      NQ1 = NQ - 1
      NQ2 = NQ + 1
C
C     TOTAL NO. OF FLUID POINTS.
      NTPF = 0
      DO 15 K=2,NCN
   15 NTPF = NTPF + NPINTS(K)
      NTPF = (NTPF*NCIR) + NPINTS(1)
C-    POINTS IN THE FIRST ANNULAR SPACE
      NP1 = NPINTS(1)
```

```
      NP2 = NPINTS(2)
      READ (5,1002) (XO(I) ,I=1,NP1)
      READ (5,1002) (XOI(I),I=1,NP2)
C
      Y1 = 0.0
      XB1 = R - SQRT(R**2-Y1**2)
      W1 = 0.0
      W1 = AI(1) + AI(2)*Y1 + AI(3)*Y1**2 + AI(4)*Y1**3 + AI(5)*Y1**4
      XT1 = R + SQRT(R**2-Y1**2) - W1
C
      Y2 = RB(1)
      XB2 = R - SQRT(R**2-Y2**2)
      W2 = 0.0
      W2 = AI(1) + AI(2)*Y2 + AI(3)*Y2**2 + AI(4)*Y2**3 + AI(5)*Y2**4
      XT2 = R + SQRT(R**2-Y2**2) - W2
C
      XO (1) = XB1
      XO (NP1) = XT1
      XOI(1) = XB2
      XOI(NP2) = XT2
C
      NDPB = 1
C-    FIRST POINT OF THE MODEL
      ND = ND + 1
      JDOF(1,1) = ND
      EUL (1,3) = (PI/2.0) / RADDEG
C-    ALL OTHERS IN THE FIRST BAY.
      NDP = 1
      DO 60 I=1,NP2
      THETA = 0.0
      DO 80 J=1,NCIR
      NDP = NDP + 1
      XYZ(NDP,1) = XOI(I)
      XYZ(NDP,2) = RB(1) * COS(THETA)
      XYZ(NDP,3) = RB(1) * SIN(THETA)
      IF (I.NE.NP INTS(2)) GO TO 65
      EUL(NDP,1) = ATAN2(XYZ(NDP,3),XYZ(NDP,2)) / RADDEG
      EUL(NDP,2) = 0.0
      IF (J .GT. 1) GO TO 82
      TERM = (R**2 - XYZ(NDP,2)**2)
      IF (ABS(TERM) .LT. EPS) DEG = -90.0
      IF (ABS(TERM) .LT. EPS) GO TO 82
      TERM1 = (1.0/2.0) * (R**2 -XYZ(NDP,2)**2) ** (-0.5) *
     *        (-2.0*XYZ(NDP,2))
     *          - (AI(2) + 2.0*AI(3)*XYZ(NDP,2) + 3.0*AI(4)*XYZ(NDP,2)**2
     *          + 4.0*AI(5)*XYZ(NDP,2)**3)
      DEG = ATAN(TERM1) / RADDEG
   82 CONTINUE
      EUL(NDP,3) = -90.0 - DEG
C
      NDPB = NDPB + 1
      XYZB(NDPB,1) = XYZ(NDP,1)
      XYZB(NDPB,2) = XYZ(NDP,2)
      XYZB(NDPB,3) = XYZ(NDP,3)
      EULB(NDPB,1) = EUL(NDP,1)
      EULB(NDPB,2) = EUL(NDP,2)
      EULB(NDPB,3) = EUL(NDP,3)
```

```
   65 CONTINUE
      IF (I.NE.1) GO TO 66
      EUL(NDP,1) = ATAN2(XYZ(NDP,3),XYZ(NDP,2)) / RADDEG
      EUL(NDP,3) = -ASIN((2.0*(XYZ(NDP,1)-R)) /
     1             (4.0*(XYZ(NDP,1)-R)**2 + 4.0*XYZ(NDP,3)**2 +
     2             4.0*XYZ(NDP,2)**2)**0.5) / RADDEG
      IF (J .GT. NQ2) GO TO 79
      IF (J.EQ.1 .OR. J.EQ.NQ2) GO TO 76
      ND = ND + 1
      JDOF(NDP,1) = ND
      ND = ND + 1
      JDOF(NDP,3) = ND
      GO TO 79
   76 CONTINUE
      IF (J . EQ. 1) KD = 1
      IF (J .EQ. NQ2) KD = 3
      ND = ND + 1
      JDOF(NDP,KD) = ND
      GO TO 79
   66 CONTINUE
      IF (J .GT. NQ2) GO TO 79
      IF (J.EQ.1 .OR. J.EQ.NQ2) GO TO 75
      ND = ND + 1
      JDOF(NDP,1) = ND
      ND = ND + 1
      JDOF(NDP,2) = ND
      ND = ND + 1
      JDOF(NDP,3) = ND
      GO TO 79
   75 CONTINUE
      IF (J .NE. 1) GO TO 77
      ND = ND + 1
      JDOF(NDP,1) = ND
      ND = ND + 1
      JDOF(NDP,2) = ND
      GO TO 79
   77 CONTINUE
      IF (I .NE. NP2) KD = 2
      IF (I .EQ. NP2) KD = 3
      ND = ND + 1
      JDOF(NDP,KD) = ND
   79 CONTINUE
      THETA = THETA + THETAD
   80 CONTINUE
      IF (I .EQ. 1 .OR. I .EQ. NP2) GO TO 1082
      NDP = NDP - NCIR
      N1 = NDP
      N2 = NDP + NDV2 + 2
C
      N1 = N1 + 1
      N2 = N2 - 1
      JDOF(N2,1) = -JDOF(N1,1)
      JDOF(N2,2) =  JDOF(N1,2)
C
      DO 81 J = 1,NQ1
      N1 = N1 + 1
      N2 = N2 - 1
```

```
C
      JDOF(N2,1) = -JDOF(N1,1)
      JDOF(N2,2) =  JDOF(N1,2)
      JDOF(N2,3) = -JDOF(N1,3)
   81 CONTINUE
      GO TO 1084
 1082 CONTINUE
C
      NDP = NDP - NCIR
      N1 = NDP
      N2 = NDP + NDV2 + 2
C
      N1 = N1 + 1
      N2 = N2 - 1
      JDOF(N2,1) = -JDOF(N1,1)
      JDOF(N2,2) = -JDOF(N1,2)
C
      DO 1083 J = 1,NQ1
      N1 = N1 + 1
      N2 = N2 - 1
C
      JDOF(N2,1) = -JDOF(N1,1)
      JDOF(N2,2) = -JDOF(N1,2)
      JDOF(N2,3) =  JDOF(N1,3)
 1083 CONTINUE
C
 1084 CONTINUE
C
      NDP = NDP + NCIR
C-    CENTERLINE POINTS
      NDP = NDP + 1
      XYZ(NDP,1) = XO(I+1)
      ND = ND + 1
      JDOF(NDP,2) = ND
   60 CONTINUE
C--   FOR THE TOP POINT (TOUCHING BLADDER)
      JDOF(NDP,1) = JDOF(NDP,2)
      JDOF(NDP,2) = 0
      XYZB(1,1) = XYZ(NDP,1)
      EULB(1,3) = (-PI/2.0) / RADDEG
C
      EUL(NDP,3) = EULB(1,3)
C-    ALL OTHER NODE POINTS STARTING FROM SECOND BAY OUTWARDS.
C
      DO 200 II=2,NRBAY
      NPI = NPINTS(II+1)
      READ (5,1002) (XOI(K),K=1,NPI)
      Y = RB(II)
      XB = R - SQRT(R**2-Y**2)
      W = 0.0
      W = AI(1) + AI(2)*Y + AI(3)*Y**2 + AI(4)*Y**3 + AI(5)*Y**4
      XT = R + SQRT(R**2-Y**2) - W
      XOI(1) = XB
      XOI(NPI) = XT
C
      DO 120 I=1,NPI
      THETA = 0.0
```

```
      DO 140 J=1,NCIR
      NDP = NDP + 1
      XYZ(NDP,1) = XOI(I)
      XYZ(NDP,2) = RB(II)*COS(THETA)
      XYZ(NDP,3) = RB(II)*SIN(THETA)
      IF (I.NE.NPINTS(II+1) ) GO TO 125
      EUL(NDP,1) = ATAN2(XYZ(NDP,3),XYZ(NDP,2)) / RADDEG
      EUL(NDP,2) = 0.0
      IF (J .GT. 1) GO TO 122
      TERM = (R**2 - XYZ(NDP,2)**2)
      IF (ABS(TERM) .LT. EPS) DEG = -90.0
      IF (ABS(TERM) .LT. EPS) GO TO 122
      TERM1 = (1.0/2.0) * (R**2 -XYZ(NDP,2)**2) ** (-0.5) *
     *       (-2.0*XYZ(NDP,2))
     *         - (AI(2) + 2.0*AI(3)*XYZ(NDP,2) + 3.0*AI(4)*XYZ(NDP,2)**2
     *         + 4.0*AI(5)*XYZ(NDP,2)**3)
      DEG = ATAN(TERM1) / RADDEG
  122 CONTINUE
      EUL(NDP,3) = -90.0 - DEG
C
      NDPB = NDPB + 1
      XYZB(NDPB,1) = XYZ(NDP,1)
      XYZB(NDPB,2) = XYZ(NDP,2)
      XYZB(NDPB,3) = XYZ(NDP,3)
      EULB(NDPB,1) = EUL(NDP,1)
      EULB(NDPB,2) = EUL(NDP,2)
      EULB(NDPB,3) = EUL(NDP,3)
  125 CONTINUE
      IF (I.NE.1) GO TO 126
      EUL(NDP,1) = ATAN2(XYZ(NDP,3),XYZ(NDP,2)) / RADDEG
      EUL(NDP,3) = -ASIN((2.0*(XYZ(NDP,1)-R)) /
     1                (4.0*(XYZ(NDP,1)-R)**2 + 4.0*XYZ(NDP,3)**2 +
     2                4.0*XYZ(NDP,2)**2)**0.5) / RADDEG
      IF (J .GT. NQ2) GO TO 139
      IF (J.EQ.1 .OR. J.EQ.NQ2) GO TO 136
      ND = ND + 1
      JDOF(NDP,1) = ND
      ND = ND + 1
      JDOF(NDP,3) = ND
      GO TO 139
  136 CONTINUE
      IF (J . EQ. 1) KD = 1
      IF (J .EQ. NQ2) KD = 3
      ND = ND + 1
      JDOF(NDP,KD) = ND
      GO TO 139
  126 CONTINUE
      IF (J .GT. NQ2) GO TO 139
      IF (J.EQ.1 .OR. J.EQ.NQ2) GO TO 135
      ND = ND + 1
      JDOF(NDP,1) = ND
      ND = ND + 1
      JDOF(NDP,2) = ND
      ND = ND + 1
      JDOF(NDP,3) = ND
      GO TO 139
  135 CONTINUE
```

```
      IF (J .NE. 1) GO TO 137
      ND = ND + 1
      JDOF(NDP,1) = ND
      ND = ND + 1
      JDOF(NDP,2) = ND
      GO TO 139
  137 CONTINUE
      IF (I .NE. NPI) KD = 2
      IF (I .EQ. NPI) KD = 3
      ND = ND + 1
      JDOF(NDP,KD) = ND
  139 CONTINUE
      THETA = THETA + THETAD
  140 CONTINUE
      IF (I .EQ. 1 .OR. I .EQ. NPI) GO TO 1122
      NDP = NDP - NCIR
      N1 = NDP
      N2 = NDP + NDV2 + 2
C
      N1 = N1 + 1
      N2 = N2 - 1
      JDOF(N2,1) = -JDOF(N1,1)
      JDOF(N2,2) =  JDOF(N1,2)
C
      DO 141 J = 1,NQ1
      N1 = N1 + 1
      N2 = N2 - 1
C
      JDOF(N2,1) = -JDOF(N1,1)
      JDOF(N2,2) =  JDOF(N1,2)
      JDOF(N2,3) = -JDOF(N1,3)
  141 CONTINUE
      GO TO 1124
 1122 CONTINUE
C
      NDP = NDP - NCIR
      N1 = NDP
      N2 = NDP + NDV2 + 2
C
      N1 = N1 + 1
      N2 = N2 - 1
      JDOF(N2,1) = -JDOF(N1,1)
      JDOF(N2,2) = -JDOF(N1,2)
C
      DO 1123 J = 1,NQ1
      N1 = N1 + 1
      N2 = N2 - 1
C
      JDOF(N2,1) = -JDOF(N1,1)
      JDOF(N2,2) = -JDOF(N1,2)
      JDOF(N2,3) =  JDOF(N1,3)
 1123 CONTINUE
C
 1124 CONTINUE
C
      N1 = N1 + 1
      NDP = NDP + NCIR
```

```
  120 CONTINUE
  200 CONTINUE
C
C
      CALL WRITE (XYZ,NDP,3,3HXYZ,KX)
      CALL WRITE (EUL,NDP,3,5HEULER,KX)
      CALL WRITE (XYZB,NDPB,3,4HXYZB,KS)
      CALL WRITE (EULB,NDPB,3,6HEULERB,KS)
      CALL WRITIM (JDOF,NDP,6,4HJDOF,KX)
C-    BRING XYZ,XYZB AND EUL,EULB TOGETHER ALSO ADD THE JDOFB.
C
      DO 210 I=1,NDPB
      NDPA = NDP + I
      DO 220 J=1,3
      XYZ(NDPA,J) = XYZB(I,J)
  220 EUL(NDPA,J) = EULB(I,J)
  210 CONTINUE
C-    JDOF TABLE (TOP CENTER POINT)
      NDP = NDP + 1
      NCNR = (NPINTS (2))*NCIR + NPINTS (1)
      ND = ND + 1
      JDOF(NDP,1) = ND
C-    ALL OTHER POINTS ON THE BLADDER (ON THE FIRST BAY)
      NCNR = NCNR - NCIR - 1
      DO 230 I=1,NCIR
      NCNR = NCNR + 1
      NDP = NDP + 1
      JDOF(NDP,2) = JDOF(NCNR,2)
      IF (I .GT. NQ2) GO TO 230
      IF (I.EQ.1 .OR. I.EQ.NQ2) GO TO 225
      ND = ND + 1
      JDOF(NDP,1) = ND
      ND = ND + 1
      JDOF(NDP,3) = ND
      GO TO 230
  225 CONTINUE
      IF (I .EQ. 1) ND = ND + 1
      IF (I .EQ. 1) JDOF(NDP,1) = ND
      IF (I .EQ. NQ2) ND = ND + 1
      IF (I .EQ. NQ2) JDOF(NDP,3) = ND
  230 CONTINUE
      NCNR = NCNR + 1 - NCIR
      NDP = NDP - NCIR
      N1 = NDP
      N2 = NDP + NDV2 + 2
C
      N1 = N1 + 1
      N2 = N2 - 1
      JDOF(N2,1) = -JDOF(N1,1)
      JDOF(N2,2) = -JDOF(N1,2)
C
      DO 231 I = 1,NQ1
      N1 = N1 + 1
      N2 = N2 - 1
C
      JDOF(N2,1) = -JDOF(N1,1)
      JDOF(N2,2) = -JDOF(N1,2)
```

```
      JDOF(N2,3) =  JDOF(N1,3)
  231 CONTINUE
C
      N1 = N1 + 1
      NDP = NDP + NCIR
C
C-    ALL OTHER BLADDER POINTS EXCEPT THE LAST ONE.
      NRBAY1 = NRBAY - 1
      DO 250 KK=2,NRBAY1
      NCNR = NCNR + NPINTS (KK+1) * NCIR
      DO 240 I=1,NCIR
      NCNR = NCNR + 1
      NDP = NDP + 1
      JDOF(NDP,2) = JDOF(NCNR,2)
      IF (I .GT. NQ2) GO TO 240
      IF (I.EQ.1 .OR. I.EQ. NQ2) GO TO 245
      ND = ND + 1
      JDOF(NDP,1) = ND
      ND = ND + 1
      JDOF(NDP,3) = ND
      GO TO 240
  245 CONTINUE
      IF (I .EQ. 1) ND = ND + 1
      IF (I .EQ. 1) JDOF(NDP,1) = ND
      IF (I .EQ. NQ2) ND = ND + 1
      IF (I .EQ. NQ2) JDOF(NDP,3) = ND
  240 CONTINUE
      NCNR = NCNR - NCIR
      NDP = NDP - NCIR
      N1 = NDP
      N2 = NDP + NDV2 + 2
C
      N1 = N1 + 1
      N2 = N2 - 1
      JDOF(N2,1) = -JDOF(N1,1)
      JDOF(N2,2) = -JDOF(N1,2)
C
      DO 241 I = 1,NQ1
      N1 = N1 + 1
      N2 = N2 - 1
C
      JDOF(N2,1) = -JDOF(N1,1)
      JDOF(N2,2) = -JDOF(N1,2)
      JDOF(N2,3) =  JDOF(N1,3)
  241 CONTINUE
C
      NDP = NDP + NCIR
  250 CONTINUE
C
C-    TAKE CARE OF THE LAST POINT WITH NO D.O.F.
      NDP = NDP + NCIR
C
C-    FOR ROTATIONAL DOF AT THE BLADDER POINTS
C
      NDP = NDP - NDPB
      NDP = NDP + 1
      DO 260 KK=1,NRBAY1
```

```
      DO 270 I=1,NQ2
      IF (I .EQ. NQ2) GO TO 264
      IF (I .NE. 1) GO TO 262
      NDP = NDP + 1
      ND = ND + 1
      JDOF(NDP,6) = ND
      GO TO 270
  262 CONTINUE
      NDP = NDP + 1
      ND = ND + 1
      JDOF(NDP,4) = ND
      ND = ND + 1
      JDOF(NDP,5) = ND
      ND = ND + 1
      JDOF(NDP,6) = ND
      GO TO 270
  264 CONTINUE
      NDP = NDP + 1
      ND = ND + 1
      JDOF(NDP,4) = ND
      ND = ND + 1
      JDOF(NDP,5) = ND
  270 CONTINUE
      NDP = NDP - NQ2
      N1 = NDP
      N2 = NDP + NDVZ + 2
C
      N1 = N1 + 1
      N2 = N2 - 1
      JDOF(N2,4) =  JDOF(N1,4)
      JDOF(N2,5) =  JDOF(N1,5)
      JDOF(N2,6) = -JDOF(N1,6)
C
      DO 271 I = 1,NQ1
      N1 = N1 + 1
      N2 = N2 - 1
C
      JDOF(N2,4) =  JDOF(N1,4)
      JDOF(N2,5) =  JDOF(N1,5)
      JDOF(N2,6) = -JDOF(N1,6)
  271 CONTINUE
C
      NDP = NDP + NCIR
  260 CONTINUE
C
C
      CALL WRITE (XYZ,NDPA,3,3HXYZ,KX)
      CALL WRITE (EUL,NDPA,3,5HEULER,KX)
      CALL WRITIM (JDOF,NDPA,6,4HJDOF,KX)
C
CCCCCC-----------------------------------------------------------------
C
      WRITE (NUTXYZ) NDPA,NCX,NDPA,NCJ,NDPA,NCX
      WRITE(NUTXYZ) ((JDOF(I,J),I=1,NDPA),J=1,NCJ)
      WRITE (NUTXYZ)  (( XYZ(I,J),I=1,NDPA),J=1,NCX)
      WRITE (NUTXYZ)  (( EUL(I,J),I=1,NDPA),J=1,NCX)
C
```

```
C-    ELEMENTS IN THE FIRST BAY
C     PENTAS
      NJ = 1
      NCOUNT = NELEMI(1) - 1
      DO 310 I=1,NCOUNT
      NJ = NJ + NCIR + 1
      N1 = NJ
      N2 = NJ + 1
      N3 = NJ
      N4 = NJ - NCIR - 1
      N5 = NJ - NCIR
      N6 = NJ - NCIR - 1
      DO 320 J=1,NCIR1
      NEL = NEL + 1
      N2 = N2 + 1
      N3 = N3 + 1
      N5 = N5 + 1
      N6 = N6 + 1
      ICONF(NEL,1) = N1
      ICONF(NEL,2) = N2
      ICONF(NEL,3) = N3
      ICONF(NEL,4) = N4
      ICONF(NEL,5) = N5
      ICONF(NEL,6) = N6
  320 CONTINUE
  310 CONTINUE
C     TETRA (TOP)
      NJ = NJ + NCIR +1
      N4 = NJ
      N1 = NJ - NCIR -1
      N2 = N1
      NB1 = NTPF + 1
      NB2 = NB1
      DO 330 I=1,NCIR1
      NEL = NEL + 1
      N2 = N2 + 1
      N3 = N2 + 1
      ICONF(NEL,1) = N1
      ICONF(NEL,2) = N2
      ICONF(NEL,3) = N3
      ICONF(NEL,4) = N4
      NELS = NELS + 1
      ICONS(NELS,1) = N4
      ICONS(NELS,2) = N2
      ICONS(NELS,3) = N3
      NELB = NELB + 1
      NB2 = NB2 + 1
      NB3 = NB2 + 1
      ICONB(NELB,1) = NB1
      ICONB(NELB,2) = NB2
      ICONB(NELB,3) = NB3
  330 CONTINUE
      NJ = NJ - (NCIR+1)
      NJ = NJ - (NCIR+1) * NCOUNT
      NJ1 = NJ
      NJ2 = NPINTS (1) + (NBINTS (2)*NCIR)
C
```

```
      C-     READ IN THE BOTTOM AND TOP ELEMENTS IDENTIFICATIONS.
             READ (5,3002) ELMID1,ELMID2
      C-     BOTTOM ELEMENT OF THE SECOND BAY.
             READ (5,3002) ELMID1,ELMID2
      C
      C
             IF (ELMID1 .NE. 5HPENTA) GO TO 340
             N1 = NJ1
             N2 = NJ1 + NCIR + 1
             N3 = NJ2
             DO 350 I=1,NCIR1
             NEL = NEL + 1
             N1 = N1 + 1
             N2 = N2 + 1
             N3 = N3 + 1
             N4 = N1 + 1
             N5 = N2 + 1
             N6 = N3 + 1
             ICONF(NEL,1) = N1
             ICONF(NEL,2) = N3
             ICONF(NEL,3) = N2
             ICONF(NEL,4) = N4
             ICONF(NEL,5) = N6
             ICONF(NEL,6) = N5
         350 CONTINUE
             NJ1 = N5
             NJ2 = N6
             GO TO 370
         340 CONTINUE
             N1 = NJ1 + NCIR + 1
             N3 = NJ2 + NCIR + 1
             N5 = NJ1
             N7 = NJ2 + 1
             DO 360 I=1,NCIR1
             NEL = NEL + 1
             N1 = N1 + 1
             N2 = N1 + 1
             N3 = N3 + 1
             N4 = N3 - 1
             N5 = N5 + 1
             N6 = N5 + 1
             N7 = N7 + 1
             N8 = N7 - 1
             ICONF(NEL,1) = N1
             ICONF(NEL,2) = N2
             ICONF(NEL,3) = N3
             ICONF(NEL,4) = N4
             ICONF(NEL,5) = N5
             ICONF(NEL,6) = N6
             ICONF(NEL,7) = N7
             ICONF(NEL,8) = N8
         360 CONTINUE
             NJ1 = N2 - NCIR
             NJ2 = N3 - NCIR
         370 CONTINUE
      C-     IN BETWEEN ELEMENTS IN THE SECOND BAY (EXCLUDING TOP + BOTTOM)
      C
```

```
      NCOUNT = NELEMI(2) - 2
      N1 = NJ1 + NCIR + 1
      N3 = NJ2 + NCIR + 1
      N5 = NJ1
      N7 = NJ2 + 1
      DO 380 I=1,NCOUNT
      DO 390 J=1,NCIR1
      NEL = NEL + 1
      N1 = N1 + 1
      N2 = N1 + 1
      N3 = N3 + 1
      N4 = N3 - 1
      N5 = N5 + 1
      N6 = N5 + 1
      N7 = N7 + 1
      N8 = N7 - 1
      ICONF(NEL,1) = N1
      ICONF(NEL,2) = N2
      ICONF(NEL,3) = N3
      ICONF(NEL,4) = N4
      ICONF(NEL,5) = N5
      ICONF(NEL,6) = N6
      ICONF(NEL,7) = N7
      ICONF(NEL,8) = N8
  390 CONTINUE
      N1 = N1 - NCIR1 + NCIR + 1
      N3 = N3 - NCIR1 + NCIR
      N5 = N5 - NCIR1 + NCIR + 1
      N7 = N7 - NCIR1 + NCIR
      NJ1 = N2 - NCIR
      NJ2 = N3 - NCIR - 1
  380 CONTINUE
C
C-    FOR THE LAST ELEMENT
      NKOUNT = NPINTS(2)
      IF (ELMID1 .EQ. 5HPENTA) NKOUNT  = NKOUNT  - 1
      IF (NKOUNT  .GT. NPINTS (3)) NJ3 = NJ1 + NCIR + 1
      IF (NKOUNT  .LT. NPINTS (3)) NJ3 = NJ2 + NCIR
      IF (NKOUNT  .EQ. NPINTS (3)) GO TO 400
C
      NB1 = NTPF + 1 + 1
      NB3 = NTPF + NCIR + 1
      N1 = NJ1
      N2 = NJ2
      N3 = NJ3
      DO 410 I=1,NCIR1
      NEL = NEL + 1
      N1 = N1 + 1
      N2 = N2 + 1
      N3 = N3 + 1
      N4 = N1 + 1
      N5 = N2 + 1
      N6 = N3 + 1
      ICONF(NEL,1) = N1
      ICONF(NEL,2) = N3
      ICONF(NEL,3) = N2
      ICONF(NEL,4) = N4
```

```
      ICONF(NEL,5) = N6
      ICONF(NEL,6) = N5
      IF (NKOUNT  .LT. NPINTS (3)) GO TO 395
      NS1 = N3
      NS2 = N2
      NS3 = N5
      NS4 = N6
      NB1 = NB1 + 1
      NB2 = NB1 - 1
      NB3 = NB3 + 1
      NB4 = NB3 + 1
      GO TO 397
  395 CONTINUE
      NS1 = N1
      NS2 = N3
      NS3 = N6
      NS4 = N4
      NB1 = NB1 + 1
      NB2 = NB1 - 1
      NB3 = NB3 + 1
      NB4 = NB3 + 1
  397 CONTINUE
      NELB = NELB + 1
      ICONB(NELB,1) = NB1
      ICONB(NELB,2) = NB2
      ICONB(NELB,3) = NB3
      ICONB(NELB,4) = NB4
      NELS = NELS + 1
      ICONS(NELS,1) = NS1
      ICONS(NELS,2) = NS2
      ICONS(NELS,3) = NS3
      ICONS(NELS,4) = NS4
  410 CONTINUE
      IF (NKOUNT  .GT. NPINTS (3)) NJ2 = NJ2
      IF (NKOUNT  .LT. NPINTS (3)) NJ2 = NJ3
      GO TO 430
  400 CONTINUE
        NJ3 = NJ2 + NCIR
        NJ4 = NJ1 + NCIR+1
        N1 = NJ4
        N4 = NJ3
        N5 = NJ1

      N8 = NJ2
      DO 420 I=1,NCIR1
      NEL = NEL + 1
      N1 = N1 + 1
      N2 = N1 + 1
      N4 = N4 + 1
      N3 = N4 + 1
      N5 = N5 + 1
      N6 = N5 + 1
      N8 = N8 + 1
      N7 = N8 + 1
      ICONF(NEL,1) = N1
      ICONF(NEL,2) = N2
      ICONF(NEL,3) = N3
```

```
      ICONF(NEL,4) = N4
      ICONF(NEL,5) = N5
      ICONF(NEL,6) = N6
      ICONF(NEL,7) = N7
      ICONF(NEL,8) = N8
      NS1 = N2
      NS2 = N1
      NS3 = N4
      NS4 = N3
      NELS = NELS + 1
      ICONS(NELS,1) = NS1
      ICONS(NELS,2) = NS2
      ICONS(NELS,3) = NS3
      ICONS(NELS,4) = NS4
      NELB = NELB + 1
      NB1 = NB1 + 1
      NB2 = NB1 + 1
      NB3 = NB3 + 1
      NB4 = NB3 + 1
      ICONB(NELB,1) = NB1
      ICONB(NELB,2) = NB2
      ICONB(NELB,3) = NB3
      ICONB(NELB,4) = NB4
  420 CONTINUE
      NJ2 = NJ3
      NJ3 = NJ4
  430 CONTINUE
C
C-    ALL OTHER ELEMENTS FROM THIRD TO LAST BUT ONE BAY.
C
      NBT1 = NTPF + 1 + 1
      NBT3 = NTPF + NCIR + 1
      NB1 = NBT1
      NB3 = NBT3
      NJ1 = NJ3 + NCIR + 1
      NJ2 = NJ2 + NCIR
      DO 500 II=3,NRBAY1
      NB1 = NB1 + NCIR
      NB3 = NB3 + NCIR
      READ (5,3002) ELMID1,ELMID2
C-    FIRST (BOTTOM)ELEMENT OF EACH
      IF (ELMID1 .NE. 5HPENTA) GO TO 510
      NJJ1 = NJ1
      NJJ2 = NJ2
      NJJ3 = NJ1 + NCIR
      N1 = NJJ1
      N2 = NJJ2
      N3 = NJJ3
      DO 520 I=1,NCIR1
      NEL = NEL + 1
      N1 = N1 + 1
      N2 = N2 + 1
      N3 = N3 + 1
      N4 = N1 + 1
      N5 = N2 + 1
      N6 = N3 + 1
      ICONF(NEL,1) = N1
```

```
      ICONF(NEL,2) = N3
      ICONF(NEL,3) = N2
      ICONF(NEL,4) = N4
      ICONF(NEL,5) = N6
      ICONF(NEL,6) = N5
  520 CONTINUE
      NJC1 = NJJ3
      NJC2 = NJJ2
      GO TO 540
  510 CONTINUE
      NJJ1 = NJ1
      NJJ2 = NJ2
      NJJ3 = NJ2 + NCIR
      NJJ4 = NJ1 + NCIR
      N1 = NJJ4
      N4 = NJJ3
      N5 = NJJ1
      N8 = NJJ2
      DO 530 I=1,NCIR1
      NEL = NEL + 1
      N1 = N1 + 1
      N2 = N1 + 1
      N4 = N4 + 1
      N3 = N4 + 1
      N5 = N5 + 1
      N6 = N5 + 1
      N8 = N8 + 1
      N7 = N8 + 1
      ICONF(NEL,1) = N1
      ICONF(NEL,2) = N2
      ICONF(NEL,3) = N3
      ICONF(NEL,4) = N4
      ICONF(NEL,5) = N5
      ICONF(NEL,6) = N6
      ICONF(NEL,7) = N7
      ICONF(NEL,8) = N8
  530 CONTINUE
      NJC1 = NJJ4
      NJC2 = NJJ3
  540 CONTINUE
C
      IF ( NELEMI(II) .LE. 2) GO TO 1050
C-    IN BETWEEN ELEMENTS ELEMENTS OF EACH
      NCOUNT = NELEMI(II) - 2
      DO 550 I=1,NCOUNT
      NJC3 = NJC2 + NCIR
      NJC4 = NJC1 + NCIR
      N1 = NJC4
      N4 = NJC3
      N5 = NJC1
      N8= NJC2
      DO 560 J=1,NCIR1
      NEL = NEL + 1
      N1 = N1 + 1
      N2 = N1 + 1
      N4 = N4 + 1
      N3 = N4 + 1
```

```
      N5 = N5 + 1
      N6 = N5 + 1
      N8 = N8 + 1
      N7 = N8 + 1
      ICONF(NEL,1) = N1
      ICONF(NEL,2) = N2
      ICONF(NEL,3) = N3
      ICONF(NEL,4) = N4
      ICONF(NEL,5) = N5
      ICONF(NEL,6) = N6
      ICONF(NEL,7) = N7
      ICONF(NEL,8) = N8
  560 CONTINUE
      NJC1 = NJC4
      NJC2 = NJC3
  550 CONTINUE
C
 1050 CONTINUE
      IF ( NELEMI(II) .LT. 2) GO TO 580
C-    FOR THE LAST ELEMENT IN EACH BAY.
C
      NKOUNT = NPINTS(II)
      IF (ELMID1 .EQ. SHPENTA) NKOUNT  = NKOUNT  - 1
      IF (NKOUNT  .GT. NPINTS (II+1)) NJC3 = NJC1 + NCIR
      IF (NKOUNT  .LT. NPINTS (II+1)) NJC3 = NJC2 + NCIR
      IF (NKOUNT  .EQ. NPINTS (II+1)) GO TO 565
      N1 = NJC1
      N2 = NJC2
      N3 = NJC3
      DO 570 I=1,NCIR1
      NEL = NEL + 1
      N1 = N1 + 1
      N2 = N2 + 1
      N3 = N3 + 1
      N4 = N1 + 1
      N5 = N2 + 1
      N6 = N3 + 1
      ICONF(NEL,1) = N1
      ICONF(NEL,2) = N3
      ICONF(NEL,3) = N2
      ICONF(NEL,4) = N4
      ICONF(NEL,5) = N6
      ICONF(NEL,6) = N5
      IF (NKOUNT  .LT. NPINTS (II+1)) GO TO 555
      NS1 = N3
      NS2 = N2
      NS3 = N5
      NS4 = N6
      GO TO 557
  555 CONTINUE
      NS1 = N1
      NS2 = N3
      NS3 = N6
      NS4 = N4
  557 CONTINUE
      NELS = NELS + 1
      ICONS(NELS,1) = NS1
```

```
      ICONS(NELS,2) = NS2
      ICONS(NELS,3) = NS3
      ICONS(NELS,4) = NS4
      NELB = NELB + 1
      NB1 = NB1 + 1
      NB2 = NB1 - 1
      NB3 = NB3 + 1
      NB4 = NB3 + 1
      ICONB(NELB,1) = NB1
      ICONB(NELB,2) = NB2
      ICONB(NELB,3) = NB3
      ICONB(NELB,4) = NB4
  570 CONTINUE
      IF (NKOUNT  .GT. NPINTS (II+1)) NJC1 = NJC3
      IF (NKOUNT  .LT. NPINTS (II+1)) NJC2 = NJC3
      GO TO 580
  565 CONTINUE
      NJC3 = NJC2 + NCIR
      NJC4 = NJC1 + NCIR
      N1 = NJC4
      N4 = NJC3
      N5 = NJC1
      N8 = NJC2
      DO 590 I=1,NCIR1
      NEL = NEL + 1
      N1 = N1 + 1
      N2 = N1 + 1
      N4 = N4 + 1
      N3 = N4 + 1
      N5 = N5 + 1
      N6 = N5 + 1
      N8 = N8 + 1
      N7 = N8 + 1
      ICONF(NEL,1) = N1
      ICONF(NEL,2) = N2
      ICONF(NEL,3) = N3
      ICONF(NEL,4) = N4
      ICONF(NEL,5) = N5
      ICONF(NEL,6) = N6
      ICONF(NEL,7) = N7
      ICONF(NEL,8) = N8
      NELS = NELS + 1
      NS1 = N1
      NS2 = N4
      NS3 = N4 + 1
      NS4 = N1 + 1
      ICONS(NELS,1) = NS1
      ICONS(NELS,2) = NS2
      ICONS(NELS,3) = NS3
      ICONS(NELS,4) = NS4
      NELB = NELB + 1
      NB1 = NB1 + 1
      NB2 = NB1 - 1
      NB3 = NB3 + 1
      NB4 = NB3 + 1
      ICONB(NELB,1) = NB1
      ICONB(NELB,2) = NB2
```

```
      ICONB(NELB,3) = NB3
      ICONB(NELB,4) = NB4
  590 CONTINUE
      NJC1 = NJC4
      NJC2 = NJC3
  580 CONTINUE
      NB1 = NB1 - NCIR1
      NB3 = NB3 - NCIR1
      NJ1 = NJ1 + NCIR*NPINTS (II)
      NJ2 = NJ2 + NCIR*NPINTS (II+1)
  500 CONTINUE
C-    FOR THE LAST (ONE ELEMENT) BAY.
      NJ3 = NJ2- NCIR
      N1 = NJ1
      N2 = NJ2
      N3 = NJ3
      NB1 = NB1 + NCIR
      NB3 = NB3 + NCIR
      DO 610 I=1,NCIR1
      NEL = NEL + 1
      N1 = N1 + 1
      N2 = N2 + 1
      N3 = N3 + 1
      N4 = N1 + 1
      N5 = N2 + 1
      N6 = N3 + 1
      ICONF(NEL,1) = N1
      ICONF(NEL,2) = N3
      ICONF(NEL,3) = N2
      ICONF(NEL,4) = N4
      ICONF(NEL,5) = N6
      ICONF(NEL,6) = N5
      NELS = NELS + 1
      NS1 = N3
      NS2 = N2
      NS3 = N5
      NS4 = N6
      ICONS(NELS,1) = NS1
      ICONS(NELS,2) = NS2
      ICONS(NELS,3) = NS3
      ICONS(NELS,4) = NS4
      NELB = NELB + 1
      NB1 = NB1 + 1
      NB2 = NB1 - 1
      NB3 = NB3 + 1
      NB4 = NB3 + 1
      ICONB(NELB,1) = NB1
      ICONB(NELB,2) = NB2
      ICONB(NELB,3) = NB3
      ICONB(NELB,4) = NB4
  610 CONTINUE
      CALL WRITIM (ICONF,NEL,8,4HJFLD,1500)
C-----------------------------------------------------------------------
C
C-    WRITE FLUID PROPERTY ON TAPE (INTERIOR)
      WRITE (NUTEL,3001) AN6(2)
      WRITE (NUTEL,3002) AN6(7),AN6(9),AN6(4),AN6(11),AN6(4)
```

```
      WRITE (NUTEL,3003) ANS(1),RHOF,ANS(2),BLKM
C-    WRITE FLUID ELEMENTS ON TAPE
      DO 405 L=1,NEL
  405 WRITE (NUTEL,3006) L, (ICONF(L,J),J=1,8)
      WRITE (NUTEL,3006) (INTG(J),J=1,2)
C
C-------------------------------------------------------------------------
C
C-    WRITE PROPERTY OF GRAVITY ELEMENT ON TAPE
C
      CALL WRITIM (ICONS,NELS,4,4HJSUR,KS)
      WRITE (NUTEL,3001) AN6(3)
      WRITE (NUTEL,3002) AN6(4),AN6(9),AN6(4),AN6(4),AN6(4)
      WRITE (NUTEL,3003) ANS(1),RHOF
      WRITE (NUTEL,3003) ANS(3),GVX,ANS(4),GVY,ANS(5),GVZ
C-    WRITE GRAVITY ELEMENTS ON TAPE
      DO 710 LI = 1,NELS
      L = LI + NEL
  710 WRITE (NUTEL,3006) L, (ICONS(LI,J),J=1,4)
      WRITE (NUTEL,3006)  (INTG(J),J=1,2)
C
CCCCCC------------------------------------------------------------------
C
C-    WRITE PROPERTY OF TRIANGULAR BLADDER ELEMENT (NCIR1) ON TAPE
C
      CALL WRITIM (ICONB,NELB,4,4HJBLD,KS)
      WRITE (NUTEL,3001) AN6(5)
      WRITE (NUTEL,3002) AN6(8),AN6(9),AN6(4),AN6(4),AN6(4)
      WRITE (NUTEL,3003) ANS(1),RHOBLD,ANS(6),EBLD,ANS(7),ANUBLD
      WRITE (NUTEL,3003) ANS(8),TBLD,ANS(9),TBLD,ANS(10),TBLD
C-    WRITE BLADDER TRIANGULAR ELEMENT ON TAPE
      DO 725 LI = 1,NCIR1
      L = LI + NEL + NELS
  725 WRITE (NUTEL,3004) L, (ICONB (LI,J),J=1,3), TBLD,TBLD,TBLD
      WRITE (NUTEL,3006)    (INTG(J),J=1,2)
      IF (NELB .EQ. NCIR1) GO TO 727
C
C-    WRITE BLADDER QUAD. PROPERTY ON TAPE
      WRITE (NUTEL,3001) AN6(6)
      WRITE (NUTEL,3002) AN6(8),AN6(9),AN6(4),AN6(4),AN6(4)
      WRITE (NUTEL,3003) ANS(1),RHOBLD,ANS(6),EBLD,ANS(7),ANUBLD
      WRITE (NUTEL,3003) ANS(8),TBLD,ANS(9),TBLD,ANS(10),TBLD
C-    WRITE BLADDER QUAD. ELEMENTS ON TAPE
      DO 728 LI = NCIR,NELB
      L = LI + NEL + NELS
  728 WRITE (NUTEL,3005) L, (ICONB(LI,J),J=1,4), TBLD,TBLD,TBLD
      WRITE (NUTEL,3006)    (INTG(J),J=1,2)
  727 CONTINUE
      WRITE (NUTEL,3001) AN6(1)
C
C-------------------------------------------------------------------------
C
      RETURN
      END
```

Appendix - B1

EXPLANATION OF INPUT TO STATIC FREE SURFACE PROGRAM

Card Nos.	Input	Explanation
1	Run no., cols. 1-6; name, cols. 11-28	Three cards to satisfy subroutine "START"
2	Title 1, cols. 1-78	
3	Title 2, cols. 1-78	
4	Order of the polynomial, acceleration due to gravity (in/sec^2), ullage pressure (psi), ratio of volume fill to total volume	Format (I5,3D17.8)
5	'STOP', cols. 1-4	End of data

```
      SEPS60    SINGH
------FREE SURFACE FOR MERCURY SLOSH---------------------60 PERCENT FULL
------GRAVITY FIELD--------------0.00386
--------    4    0.00386          0.10              0.60
------STOP
```

Appendix - B2

EXPLANATION OF INPUT FOR DYNAMIC ANALYSIS PROGRAM

Card Nos.	Input	Explanation
1-31		Control cards (if needed)
32	Run no., cols. 1-6; name, cols. 11-28	Three cards to satisfy subroutine "START"
33	Title 1, cols. 1-78	
34	Title 2, cols. 1-78	
35	Users comment cards	Any no. of cards, the last card must be zeros. Cols. 1-10. Subroutine "COMENT"
36	'INITIL' or 'NOINIT', cols. 1-6	To initialize or not to initialize the reserve tape
37	'GNXYZ', cols. 1-5	To call subroutine "GNXYZ3"
38	'XYZEUL', cols. 1-6	To call subroutine "XYZEU3"
39	Radius of the sphere, cols. 1-10	Format (F10.0)
40	2 x mass density, 2 x bulk modulus, of the fluid and gravity components in x, y, and z directions	Format (5F10.0)
41	2 x mass density, 2 x Young's modulus, Poisson's ratio, and thickness of the bladder	Format (4F10.0)
42	Number of circumferential points and number of radial bays in the half space model	Format (2I5)
43	Name, no. of rows, no. of cols. of the matrix	Subroutine "READ"
44	The coefficients of the polynomial	
45	assumed for the free surface	
46	Zeros	
47	Name, no. of rows, no. of cols. of the matrix	Subroutine "READ"
48, 49, 50	bay radii	
51	Zeros	

ORIGINAL PAGE IS OF POOR QUALITY

52	Name, no. of rows, no. of cols. of the matrix	Subroutine "READIM"
53	Number of points on each vertical line	
54	Zeros	
58-70	X-coord. of the points on each vertical line from inside-out	Format (8F10.0)
71-81	Identification of the first and the last element in each bay from bottom to top	Format (2(A6,4X))
82	'FINEL', cols. 1-5	Calls subroutine "FINEL"
83	'MODES', cols. 1-5	Calls subroutine "OYMODE"
84	Name, no. of rows, no. of cols. of the matrix	Matrix of assumed mode shapes (if any). Subroutine "READ"
85	Zeros	
86	Number of modes wanted	Format (10X,I5)
87	Number of modes used	Format (10X,I5)
88	Shift value for ω^2 (convergence will be about this value)	Format (10X,E17.0)
89	No. of maximum iteration allowed	Format (10X,I5)
90	'PUNCH' or 'NOPNCH', cols. 1-6	Option for punch output
91	'MECHEQ', cols. 1-6	Calls subroutine "MECHEQ" to calculate mechanical equivalent
92	Name, logical tape no., run no.	To read x, y, z locations of the points. Subroutine "READ"
93	Last joint with only 3d.o.f., last d.o.f. number	Format (2I5)
94	Name, no. of rows, no. of cols. in the matrix	Location of the reference point. Subroutine "READ"
95	Zeros	
96	Name, no. of rows, no. of cols. in the matrix	The elements of the matrix indicate which cols. of the rigid body matrix are non-zeros. Subroutine "READIM"
97	Column no. of rigid body matrix (non-zero)	
98	Zeros	
99	Name, logical tape no., run no.	Degrees of freedom matrix. Subroutine "READ"

100	Name, logical tape no., run no.	Euler angle matrix Subroutine "READ"
101	Name, logical tape no., run no.	Mass matrix Subroutine "READ"
102	Name, logical tape no., run no.	Modes matrix Subroutine "READ"
103	'PLOT', cols. 1-4	Option to plot the mode shapes (NO card, no plot)
104	Name, no. of rows, no. of cols. in the matrix	Joint numbers which are to be plotted Subroutine "READIM"
105	Node no. to be plotted	
106	Zeros	
107	Name, no. of rows, no. of cols. of the matrix	x, y, z location of the points Subroutine "READ"
108	Name, no. of rows, no. of cols. of the matrix	Euler angles of the points Subroutine "READ"
109	Name, no. of rows, no. of cols. of the matrix	Frequencies of the system Subroutine "READ"
110	Name, no. of rows, no. of cols. of the matrix	Modes of the system Subroutine "READ"
111	Name, no. of rows, no. of cols. of the matrix	d.o.f. matrix of the system Subroutine "READ"
112	'START', cols. 1-5	Preparatory to end
113	'STOP', cols. 1-4	End of data
114-118		Control cards

```
=RUN,Q M08GB1,HMMHEA8704,S-BULTM00000,30,2000/1200    BULTMAN  BIN S-190




=US*ER.BOXPCH   ,S-195-BULT.
=FREE  TPF$.
=ASG,T  TPF$,F2/1/POS/500
=ASG,T  T,U,17701
=COPY,G  T.,TPF$.
=FREE  T.
=ASG,T     1.,F40/1/POS/10
=ASG,T     2.,F40/1/POS/10
=ASG,T    11.,F40/1/POS/10
=ASG,T    12.,F40/1/POS/10
=ASG,T    13.,F40/1/POS/10
=ASG,T    14.,F40/1/POS/10
=ASG,T    15.,F40/1/POS/10
=ASG,T    16.,F40/1/POS/10
=ASG,T    17.,F40/1/POS/10
=ASG,T    21.,F40/1/POS/10
=ASG,T    22.,F40/1/POS/10
=ASG,T    23.,F40/1/POS/10
=ASG,T    24.,F40/1/POS/10
=ASG,T    25.,F40/1/POS/10
=ASG,T    26.,F40/1/POS/10
=ASG,T    27.,F40/1/POS/10
=ASG,T   28.U
=ASG,T    30.,F40/1/POS/10
=ASG,T    31.,F40/1/POS/10
=ELT,ILD  DATA
RUN140    PHILIPPUS
LATERAL SLOSH FOR SPHERICAL TANK WITH MERCURY AND BLADDER IN IT.
SPHERE RIGID----- 40 PERCENT FULL--------- G # 386.0 IN / SEC**2.
0000000000
INITIL
GNXYZ
XYZEUL
   8.00
   0.0026   2000000.0   -386.0      0.0        0.0
 0.000228    400.00          0.45     0.06
    7   12
AI        1      5
    1     1      8.426723              0.00              1.164276          -0.356903
    1     5      0.024364
0000000000
RR        1      12
    1     1      1.00                  2.00              2.75              3.30
    1     5      4.30                  5.40              6.20              6.9
    1     9      7.3                   7.5               7.9               8.0
0000000000
NPTS      1      13
    1     1      5     4     4     4     3     4     4     5     5     5     4     2     1
0000000000
NELEMI    1      12
    1     1      4     3     3     3     3     4     4     5     5     4     3     1
0000000000
```

```
      0.0         2.3       4.1       6.68      7.57
      0.06275     2.3       4.1       6.68
      0.3         2.3       4.1       5.13
      0.5         2.3       4.1       4.28
      0.71        2.3       4.1
      1.3         2.3       4.1       4.75
      2.3         4.1       4.75      7.0
      2.95        4.1       4.75      7.0       9.0
      4.1         4.75      7.0       9.0       10.2
      4.75        7.0       9.0       10.2      10.5
      5.4         7.0       9.0       10.2
      7.0         9.0
      8.00
  PENTA     TETRA
  HEXA      HEXA
  HEXA      HEXA
  HEXA      PENTA
  HEXA      PENTA
  PENTA     PENTA
  HEXA      PENTA
  PENTA     PENTA
  PENTA     PENTA
  HEXA      PENTA
  PENTA     PENTA
  FINEL
  MODES
  INMODE    1      1
  0000000000
  NW               6
  NU               6
  SHIFT     500.0
  MAXIT           15
  NOPUNCH
  MECHEQ
  XYZ            -28RUN140
     321   375
  XREF      1      3
  0000000000
  JVEC      1      6
     1      1      1      2      0      0      0      3
  0000000000
  JDOF           -28RUN140
  EUL            -28RUN140
  MASS           -28RUN140
  MODES          -28RUN140
  PLOT
  SURFAS    1      13
     1      1      33     26     55     83    104    132    160    195    230    265    293    307    314
  0000000000
  XYZ            -28RUN140
  EUL            -28RUN140
  FREQ           -28RUN140
  MODES          -28RUN140
  JDOF           -28RUN140
  START
  STOP
  =END
  =XQT   MM200G
  =ADD,PL   DATA
  =PMD,PLE
```

118 =FIN